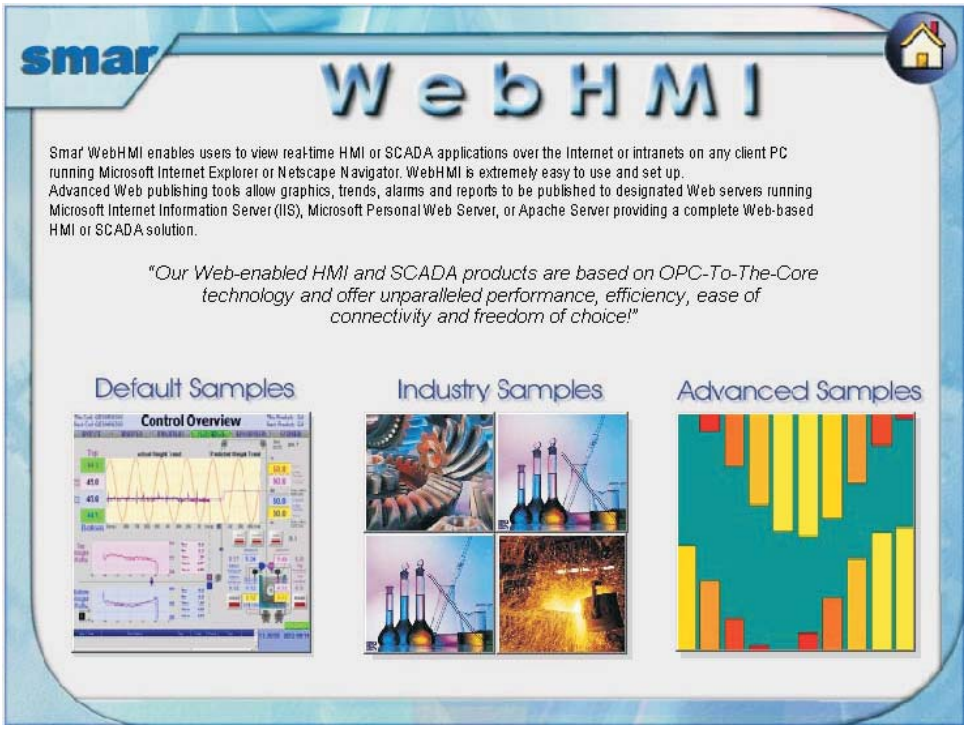
- Web HMI

Smar First in Fieldbus

Web HMI


smar
WebHMI
Smar WebHMI enables users to view real-time HMI or SCADA applications over the Internet or intranets on any client PC running Microsoft Internet Explorer or Netscape Navigator. WebHMI is extremely easy to use and set up.
Advanced Web publishing tools allow graphics, trends, alarms and reports to be published to designated Web servers running Microsoft Internet Information Server (IIS), Microsoft Personal Web Server, or Apache Server providing a complete Web-based HMI or SCADA solution.
"Our Web-enabled HMI and SCADA products are based on OPC-To-The-Core technology and offer unparalleled performance, efficiency, ease of connectivity and freedom of choice!"
Default Samples
Control Overview
Industry Samples
Advanced Samples

MAY / 06

Web HMI

VERSION 8

FOUNDATION™

PVIEWHMI ME

www.smar.com

Specifications and information are subject to change without notice.
Up-to-date address information is available on our website.

web: www.smar.com/contactus.asp

TABLE OF CONTENTS

CHAPTER 1 - INTRODUCTION 1.1
ABOUT THIS DOCUMENT 1.1
KEY FEATURES OF WEBHMI 1.1
INTRODUCTION TO WEBHMI 1.1
WEBHMI PRODUCT CONTENTS 1.2
WEBHMI ARCHITECTURE 1.2
TRACEWORX SUPPORT 1.3

CHAPTER 2 - SETTING UP THE WEB SERVER 2.1
INTRODUCTION 2.1
INSTALLING WEB SERVERS FOR WINDOWS NT AND WINDOWS 2000 ..... 2.1
INSTALLING INTERNET INFORMATION SERVER FOR WINDOWS NT 2.1
INSTALLING INTERNET INFORMATION SERVER FOR WINDOWS 2000 ..... 2.4
INSTALLING A PERSONAL WEB SERVER FOR WINDOWS NT 2.5
INSTALLING PROCESSVIEW 2.6
INSTALLING THE PROCESSVIEW LICENSE UTILITY 2.7
INSTALLING WEBHMI 2.7

CHAPTER 3 - VIEWING WEBHMI SAMPLES 3.1
INTRODUCTION TO WEBHMI SAMPLES 3.1
DEFAULT SAMPLES 3.1
GRAPHWORX SAMPLES 3.2
ALARMWORX SAMPLES 3.2
TRENDWORX SAMPLES 3.3
ACTIVEX CONTROL SAMPLES 3.3
LICENSING AND SECURITY SAMPLES 3.4
ADVANCED SAMPLES 3.4
INDUSTRY SAMPLES 3.6

CHAPTER 4 - WEB PUBLISHING WIZARD 4.1
INTRODUCTION TO HTML PUBLISHING 4.1
BASIC STEPS IN PUBLISHING GRAPHWORX DISPLAYS 4.1
DELIVERING THE NECESSARY WEB COMPONENTS TO THE CLIENT 4.2
MULTIPLE DISPLAY SUPPORT 4.2
EMBEDDED ACTIVEX CONTROL SUPPORT 4.3
USING THE WEB PUBLISHING WIZARD 4.3
STARTING THE WEB PUBLISHING WIZARD 4.3
EXPORTING A DISPLAY FILE LOCALLY 4.5
PUBLISHING A DISPLAY FILE TO A WEB SERVER 4.7
PUBLISHING CUSTOMIZATION OPTIONS 4.8
VIEWING LOCALLY EXPORTED HTML FILES 4.12
VIEWING PUBLISHED HTML FILES 4.12
WEB PUBLISHING IN PROJECTWORX 4.13
WEB PUBLISHING FROM ALARM AND TREND VIEWER FILES 4.14
GRAPHWORX-BASED HTML 4.15
ALARMWORX-BASED HTML 4.15
TRENDWORX-BASED HTML 4.16
LOCAL HTML FILES 4.16
PUBLISHED DOCUMENTS 4.17
LAUNCHING THE WEB PUBLISHING WIZARD IN PROJECTWORX 4.17
EXPORT AND PUBLISH OPTIONS IN PROJECTWORX 4.20
EXPORTING AND PUBLISHING MULTIPLE DISPLAY FILES IN PROJECTWORX 4.20
EXPORTING AND PUBLISHING INDIVIDUAL DISPLAY FILES IN PROJECTWORX 4.22
EXPORTING A DISPLAY FILE LOCALLY IN PROJECTWORX 4.23
PUBLISHING A DISPLAY FILE TO A WEB SERVER IN PROJECTWORX 4.24
PUBLISHING CUSTOMIZATION OPTIONS 4.26
PUBLISHING FILES AT A LATER TIME 4.29
WEB PUBLISHING LOG 4.31

CHAPTER 5 - WEBHMI CLIENT CONFIGURATION 5.1
WEBHMI CLIENT/SERVER ARCHITECTURE 5.1
CONFIGURING GENBROKER 5.1
CONFIGURING GENBROKER FOR THE CLIENT SIDE 5.2
EXAMPLE GENBROKER CONFIGURATION 5.3
COMMUNICATION USING A MEDIATOR NODE 5.4

CHAPTER 6 - WEBHMI SECURITY 6.1
WEBHMI SECURITY ACTIVEX 6.1
LOGGING INTO THE SECURITY SERVER 6.3
CHANGING THE SECURITY SERVER PASSWORD 6.4
VIEWING THE LOGGED USER LIST 6.4
LOGGING OUT OF THE SECURITY SERVER 6.5
SECURITY OLE AUTOMATION 6.5
LAUNCHING THE SECURITY LOGIN ACTIVEX THROUGH SCRIPTING 6.6

CHAPTER 7 - WEBHMI LICENSING 7.1
VIEWING LICENSING INFORMATION USING WEBHMI 7.1
LICENSE FOR THE WEBHMI SERVER 7.3
OBTAINING ADDITIONAL CONNECTION UNITS 7.3
LICENSING OLE AUTOMATION 7.3
LAUNCHING THE LICENSE MONITOR ACTIVEX THROUGH SCRIPTING 7.3

CHAPTER 8 - CHANGE HOST UTILITY 8.1
INTRODUCTION TO THE CHANGE HOST UTILITY 8.1
CHANGING THE DIRECTORY NAME 8.1
FINDING AND REPLACING FILES 8.1

CHAPTER 9 - WEBHMI VERSION SUPPORT 9.1
INTRODUCTION TO VERSION SUPPORT 9.1
UPDATE VERSIONS UTILITY 9.1
STARTING THE VERSIONS UTILITY 9.2

APPENDIX A - LARMWORX AND TRENDWORX WEB ACCESS A.1
INTRODUCTION A.1
WEB ACCESS FOR ALARM REPORTS A.1
WEB ACCESS SUPPORT FOR ALARM VIEWER OPERATOR COMMENTS A.2
TIPS FOR ALARMWORX REMOTE DATA ACCESS A.6
TRENDWORX REPORTING TOOL A.8

APPENDIX B - SETTING UP DCOM B.1
DCOM AS A REQUIREMENT B.1
CONFIGURING DCOM FOR WINDOWS NT B.1
CREATING A USER GROUP B.1
CONFIGURING DCOM PROPERTIES B.5
CONFIGURING DCOM FOR WINDOWS 98 ..... B.10
USING A WINDOWS 98 MACHINE AS A WEB SERVER B.10
DOMAINS, WORKGROUPS, RIGHTS TO THE SERVER NODE, AND DCOM B.15

Chapter 1

INTRODUCTION

About This Document

This document covers the ProcecssView WebHMI™ product, including:

- WebHMI installation
- Accessing and viewing WebHMI samples
- Web Publishing Wizard for publishing GraphWorX™, AlarmWorX™, and TrendWorX™ display files as Web pages
- AlarmWorX Web Access support for reports and operator comments.
- Setting up the WebHMI server
- WebHMI client configuration using the GenBroker™ Configurator
- GenBroker communications
- WebHMI licensing configuration
- WebHMI security configuration
- Change Host Utility
- WebHMI versions support and the Update Versions Utility

This document assumes you have prior knowledge of the following:

- Smar ProcessView™ components.
- HTML or Web page editing tools that can handle ActiveX controls.
- TCP/IP setup or DCOM (Distributed Component Object Model).

Note: Remote node names in ProcessView are often referenced by "\\<Node Name>." In this document, you will often see a remote OPC data source referenced by "\\<IP Address>." This convention is perfectly acceptable and simplifies the development of WebHMI Web pages.

Key Features of WebHMI

WebHMI includes the following key features:

- VBScript and JScript scripting engine (see the GraphWorX Help documentation for more information).
- WebHMI Security Login ActiveX.
- AlarmWorX Web Access support for reports and operator comments.
- TraceWorX diagnostics support.
- Web browser support for Microsoft Internet Explorer.
- Language switching support.

Introduction to WebHMI

WebHMI is a **Thin Client** Web solution that enables standard Web browsers (Microsoft® Internet Explorer) for use as real-time operator interfaces to manufacturing and factory floor applications. Based on ActiveX® technology, WebHMI provides you with a powerful and versatile approach to using the same standard HMI (Human Machine Interface) components included in ProcessView.

WebHMI delivers industry-standard, real-time OPC (OLE for Process Control) information. WebHMI likewise delivers fast, worldwide operator graphical visualization, trending and alarming information—both real-time and historical—and HTML-based reports.

Since WebHMI Web components are packaged in standard Microsoft .cab files, your server and clients can be located anywhere. Similarly, you can store .cab files anywhere on your network.

Installed and resident on one or more WebHMI servers, these components (e.g., GraphWorX, TrendWorX, or AlarmWorX) are delivered automatically, quickly, and "in the background" to a browser on the client-side machine. Since WebHMI delivers the necessary components required for performing HMI and SCADA functions (e.g., building control, manufacturing, and process monitoring), it is not necessary to have any Smar products installed on the client machines.

Ultimately, WebHMI turns a Web browser into an OPC client when the browser views Web pages located on any WebHMI server. As mentioned above, Microsoft Internet Explorer is used to view Web pages from the WebHMI server. Once a WebHMI site is up and running, you can instantly create multiple browser stations by having your clients launch the Web browser and visit the appropriate page.

Interaction between clients and the WebHMI server is made possible by Smar GenBroker™, which uses TCP/IP and SOAP/XML communication over the Internet. The GenBroker Configurator allows you to customize your client/server architecture based on your network configuration. For information about WebHMI client configuration, please see **Chapter 5.** For information about configuring GenBroker for the server side, please see the GenBroker Help documentation.

WebHMI Product Contents

Once you have installed WebHMI on a Web server, the following components are available:

- .cab files containing all necessary ActiveX components and foundation files.
- Registered copies of all WebHMI ActiveX components, which aid in the development of your own pages.
- Installation files for Microsoft Windows®.
- Several sample HTML projects, which any client can browse "out of the box."
- A search-and-replace application, which can find and replace strings in HTML files, dynamic tags in GraphWorX display files, and pen names in TrendWorX property page configuration files.

WebHMI Architecture

The architecture of WebHMI can best be explained in an illustration. **Figure 1.1** shows the WebHMI, Security, Licensing, and OPC servers on the same node. The WebHMI clients are used to view Web pages downloaded from the WebHMI server through a Web browser. GenBroker acts as a bridge that links WebHMI clients to the WebHMI server over the Internet using TCP/IP communications. For details about client/server architecture and GenBroker configuration, please see **Chapter 5** or refer to the GenBroker Help documentation.

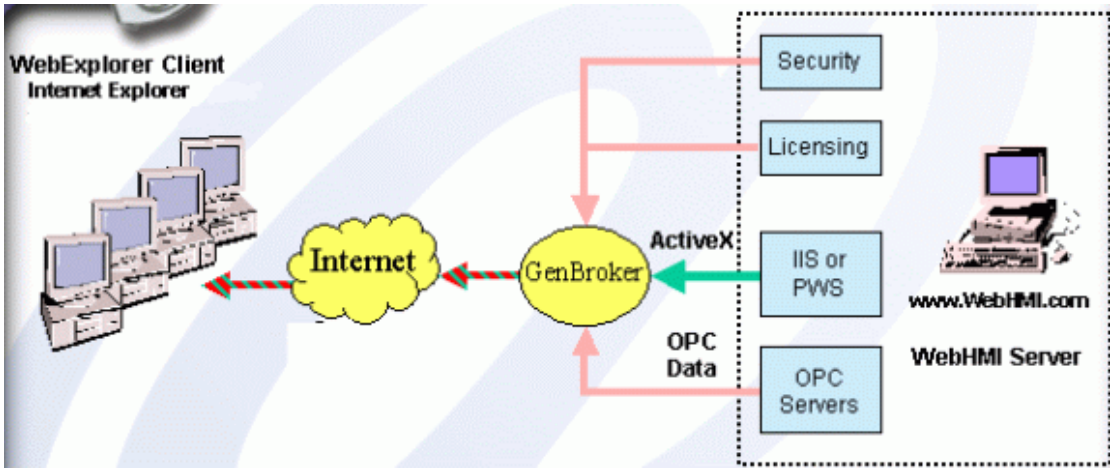


Figure 1.1. WebHMI Client/Server Architecture

Of course, you are not limited to this setup. Since ProcessView is modular, various components can be spread out over PCs with multiple networks. If you wish, you can have a single node devoted only to storing and supplying all necessary data files to be transported over HTTP; or you can have a setup in which any other node can be responsible only for supplying security information and serving Smar ' licensing. You can have as many nodes as you wish supply OPC data.

TraceWorX Support

Using a technology that has been incorporated into all Smar products, TraceWorX provides online diagnostics and tuning of applications running in the ProcessView system. TraceWorX is designed expressly for systems integrators, OEMs and customers who want to have tools for doing their own troubleshooting and diagnostics.

TraceWorX tracks the runtime activity for each ProcessView application and logs the runtime data to a log file based on user-configured trace levels. The log file provides a thorough, color-coded report detailing all activity for the application, including the time, the date, the severity level, and a description of the event or problem.

TraceWorX also features several options for reporting issues to technical support. If you are experiencing problems with any applications, the log file deployment options, such as compressing and e-mailing log files, are ideal for tracking and archiving data and sending detailed reports to technical support. Developers can use these reports to identify the source of the problems.

The **TraceWorX32.htm** file, which contains the TraceWorX Configuration Utility, is installed with ProcessView in the *Program Files*\ *Smar* *ProcessView**Bin* directory. Remote WebHMI clients can download this utility from the WebHMI Server by specifying the URL address of the **TraceWorX32.htm** file (e.g. http://www.myserver.com/WebHMI/TraceWorX32.htm). The TraceWorX Utility will track any problems on the remote client.

Note: The **View** button on the TraceWorX Configuration Utility is disabled in WebHMI.

For more information about using TraceWorX, please see the **TraceWorX32 Help.htm** file in the *Program Files**Smar**ProcessView**Bin* directory, or see the TraceWorX Help documentation on the ProcessView product CD.

Chapter 2

SETTING UP THE WEB SERVER

Introduction

In this chapter, the following topics will be covered:

- Installing Internet Information Server in Windows NT
- Installing Internet Information Server in Windows 2000
- Installing a Personal Web Server in Windows NT
- Installing ProcessView
- Installing the ProcessView License Utility
- Installing WebHMI

Before installing WebHMI, you must first install a Web server, ProcessView, and a ProcessView software license. Any Microsoft Windows operating system that meets the ProcessView minimum system requirements and that has a Web server installed is capable of serving WebHMI pages. However, it is strongly recommended that you use Microsoft Internet Information Server (IIS) because IIS supports all features of WebHMI. Other Web servers, such as Personal Web Server and Apache®, may not support all features of WebHMI.

Note: The ProcessView Web Publishing Wizard is supported only for IIS version 4.0 or above. SOAP/XML is supported only for IIS version 5.0 or above.

Installing Web Servers for Windows NT and Windows 2000

For WebHMI to work properly on a PC, you must install the following:

- For Windows NT: Either Internet Information Server (IIS) or a Personal Web Server.
- For Windows 2000 or Windows XP: Internet Information Server (IIS).

Installing Internet Information Server for Windows NT

If you run a Windows NT server, you must install an Internet Information Server (IIS) in your computer.

1. To install IIS in Windows NT, click on **Network Neighborhood** on your computer's desktop. The **Network** dialog box will appear, as shown below in **Figure 2.1.** On the **Network** dialog box, click the **Services** tab and then click the **Add** button.

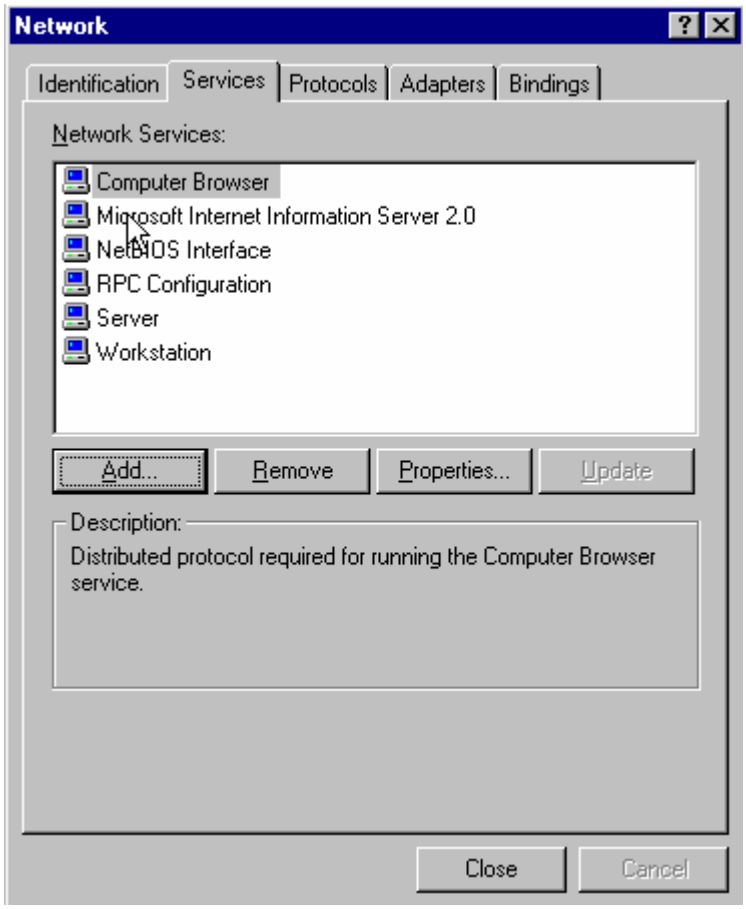


Figure 2.1. Network Dialog Box: Services Tab

2. The **Select Network Service** dialog box will appear. Select the **Network Service** (in this case Microsoft Internet Information Server), and click the **OK** button.

3. The **Internet Information Server Installation** dialog box will appear, as shown in **Figure 2.2.** This dialog box will inform you that additional files are needed for installation. In the **Installed from** field, type in the drive on which your computer runs CD-ROMs, plus ":\i386"; or click on the **Browse** button to locate the **i386** folder on your CD-ROM drive. After locating the **i386** folder, click on it to have it appear in the **Installed from** field. Insert the Windows NT CD-ROM and click on the **OK** button on the **Internet Information Server Installation** dialog box.

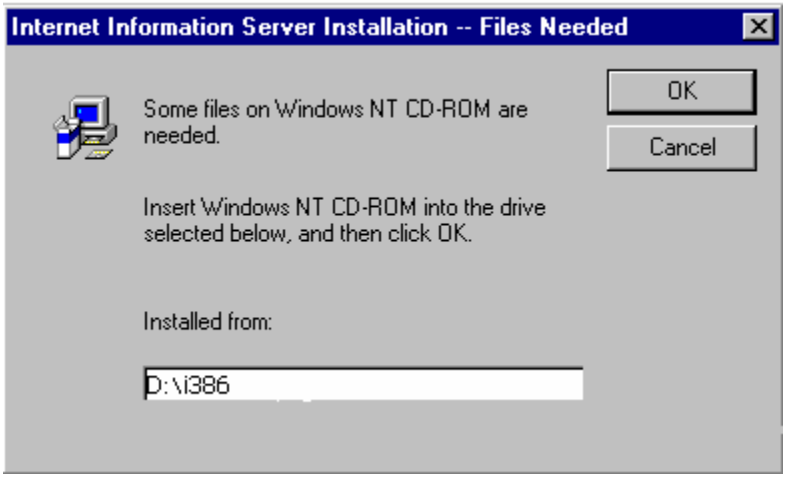

Internet Information Server Installation -- Files Needed
Some files on Windows NT CD-ROM are needed.
OK
Cancel
Insert Windows NT CD-ROM into the drive selected below, and then click OK.
Installed from:
D:\i386

Figure 2.2. Internet Information Server Installation Dialog Box

Note: The Network Service required for this setup is **Microsoft Internet Information Server 3.0,** to be released at the time of the current document's publication.

4. The **Microsoft Internet Information Server Setup** dialog box, with an **Options** window, will appear, as shown in **Figure 2.3.** In the **Options** window, be sure to click on at least the four X-boxes (**Internet Service Manager, World Wide Web Service, WWW Service Samples,** and **ODBC Drivers & Administration**). Click the **OK** button.

5. The **Publishing Directories** dialog box will appear. On the **Publishing Directories** dialog box, the default directory ("C:\InetPub\wwwroot") will appear in the **World Wide Publishing Directory** field. To use this directory, click the **OK** button.

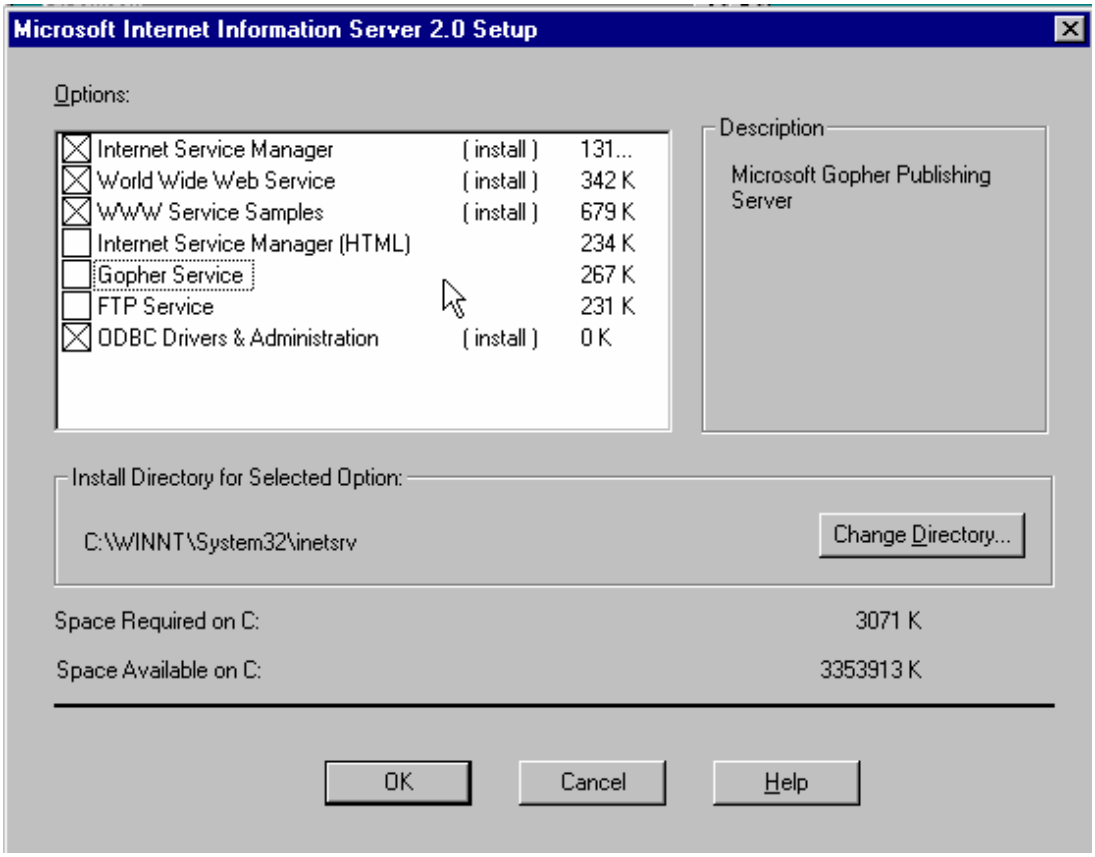

Microsoft Internet Information Server 2.0 Setup
Options:
Internet Service Manager (install) 131...
World Wide Web Service (install) 342 K
WWW Service Samples (install) 679 K
Internet Service Manager (HTML) 234 K
Gopher Service 267 K
FTP Service 231 K
ODBC Drivers & Administration (install) 0 K
Description
Microsoft Gopher Publishing Server
Install Directory for Selected Option:
C:\WINNT\System32\inetsrv
Change Directory...
Space Required on C: 3071 K
Space Available on C: 3353913 K
OK
Cancel
Help

Figure 2.3. Options Window of Microsoft Internet Information Server Dialog Box

6. If the default directory does not exist in your computer, a dialog box will appear informing you of this fact, as shown in **Figure 2.4.** If you wish to create the default directory ("C:\InetPub\wwwroot"), click **Yes.**

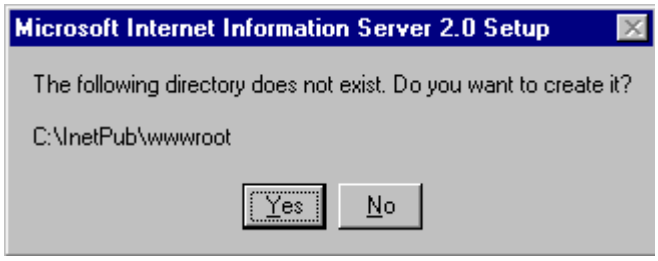

Microsoft Internet Information Server 2.0 Setup
The following directory does not exist. Do you want to create it?
C:\InetPub\wwwroot
Yes
No

Figure 2.4. Creating the C:\InetPub\wwwroot Directory

7. After the **"C:\InetPub\wwwroot"** directory has been created, the **Install Drivers** dialog box will appear, as shown below. One or more ODBC drivers will appear in the **Available ODBC Drivers** pane of the **Install Drivers** dialog box. Select the driver(s) of your choice and click **OK.**

8. The **Microsoft Internet Information Server** dialog box will appear, informing you that the setup has been completed. On the **Microsoft Internet Information Server** dialog box, click the **OK** button.

You have successfully installed Internet Information Services software on a PC with Windows NT.

Installing Internet Information Server for Windows 2000

If you have Windows 2000 with Internet Information Server (IIS), you must configure it in your computer.

1. To configure IIS in Windows 2000, enter the **Control Panel** on your computer and click on **Add/Remove Programs.** The **Add/Remove Programs** dialog box will appear. Click on the **Add/Remove Windows Components** icon in the left-hand column of the dialog box.

2. The **Windows Components Wizard** dialog box will appear, as shown in **Figure 2.5.** To add IIS, select **Internet Information Services (IIS)** and click on the check box. After you have clicked on the **IIS** check box, the **Windows Components Wizard** dialog box will change its appearance and begin configuring components. Shortly after the configuring begins, you could be notified that additional files are needed.

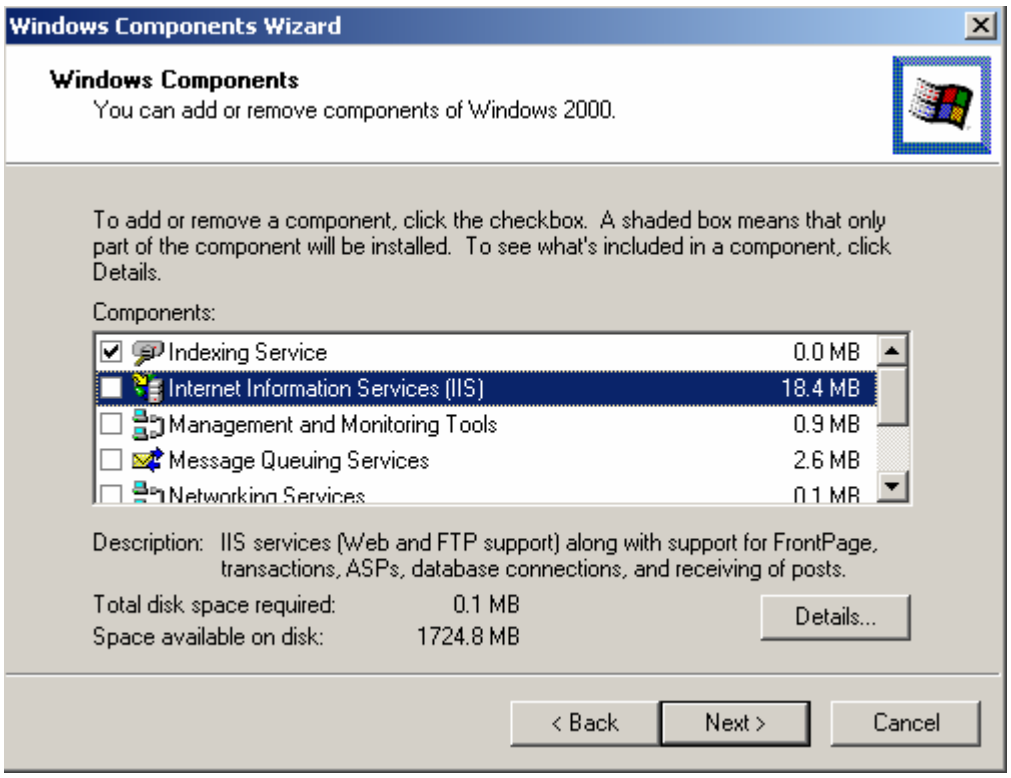

Windows Components Wizard
Windows Components
You can add or remove components of Windows 2000.
To add or remove a component, click the checkbox. A shaded box means that only part of the component will be installed. To see what's included in a component, click Details.
Components:
Indexing Service 0.0 MB
Internet Information Services (IIS) 18.4 MB
Management and Monitoring Tools 0.9 MB
Message Queuing Services 2.6 MB
Networking Services 0.1 MB
Description: IIS services (Web and FTP support) along with support for FrontPage, transactions, ASPs, database connections, and receiving of posts.
Total disk space required: 0.1 MB
Space available on disk: 1724.8 MB
Details...
< Back
Next >
Cancel

Figure 2.5. Windows Components Wizard Dialog Box

If additional files are necessary, the **Files Needed** dialog box will appear, as shown in **Figure 2.6.** In the **Copy files from** field, type in the drive on which your computer runs CD-ROMs, plus ":\i386"; or click the **Browse** button to locate the **i386** folder on your CD-ROM drive. After locating the **i386** folder, click on it to have it appear in the **Copy files from** field. Insert the Windows 2000 Professional CD-ROM and click on the **OK** button on the **Files Needed** dialog box.

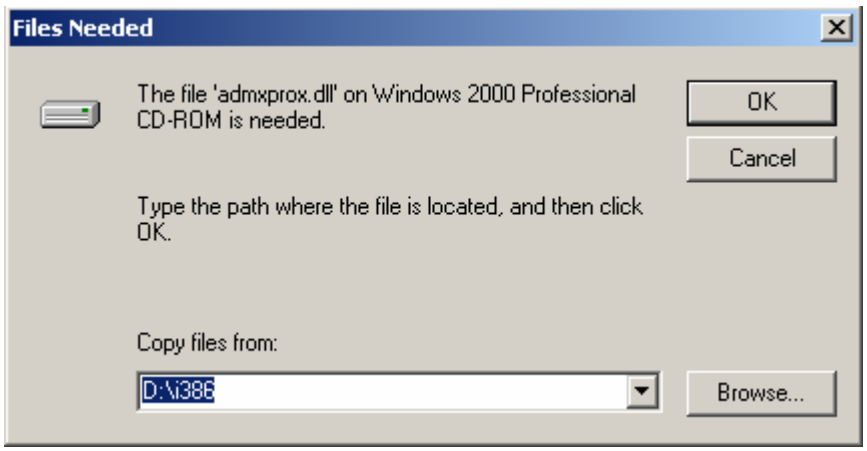


Figure 2.6. Files Needed Dialog Box

3. The **Windows Components Wizard** dialog box will reappear. The copying of files is tracked on the progress bar of the window.
4. After all of the necessary components have been installed, the **Finish** dialog box will appear. To close the **Windows Components Wizard** dialog box, click the **Finish** button.

You have successfully configured Internet Information Services software for Windows 2000.

Installing a Personal Web Server for Windows NT

If you run a Windows NT server, installing a Personal Web Server is an alternative to installing an Internet Information Server (IIS) in your computer.

1. To install a Personal Web Server, go into the **Tools** directory on the Microsoft Windows NT® 4.0 Options Pack CD-ROM and select **Tools\pws.** The **Option Pack Setup** window will appear, as shown in **Figure 2.7.**

Microsoft Windows NT 4.0 Option Pack Setup
Microsoft®
Windows NT® 4.0 Option Pack
The Windows NT 4.0 Option Pack provides enhanced Web and application development services for Windows NT Workstation 4.0.
Features include:
Personal Web Server
Transaction Server
Data Access Components
Message Queue Server Client
Internet Connection Services for RAS
Administration
Developer Components
< Back
Next >
Cancel

Figure 2.7. Microsoft Windows NT 4.0 Option Pack Setup

2. Click the **Next** button. The **End-User License Agreement** dialog box will appear. Read the conditions in the License Agreement. If you accept the conditions, click the **Accept** button.

3. The dialog box through which you choose the components to be installed on your Personal Web Server will appear, as shown in **Figure 2.8.** It is recommended that you click the **Typical** button. This configuration includes all of the minimum components, plus basic documentation and additional components that will allow you to build and deploy Web applications.

4. After you click on the **Typical** button, the dialog box with the default Web publishing home directory in the **WWW Service** field will appear. If you wish to install the default directory as your home directory, click the **Next** button.

5. The **Completing Installation** dialog box will appear. When the installation has been completed, the final dialog box will appear. Installation of a Personal Web Server for Windows NT is complete. Click the **Finish** button to close out of the installation process.

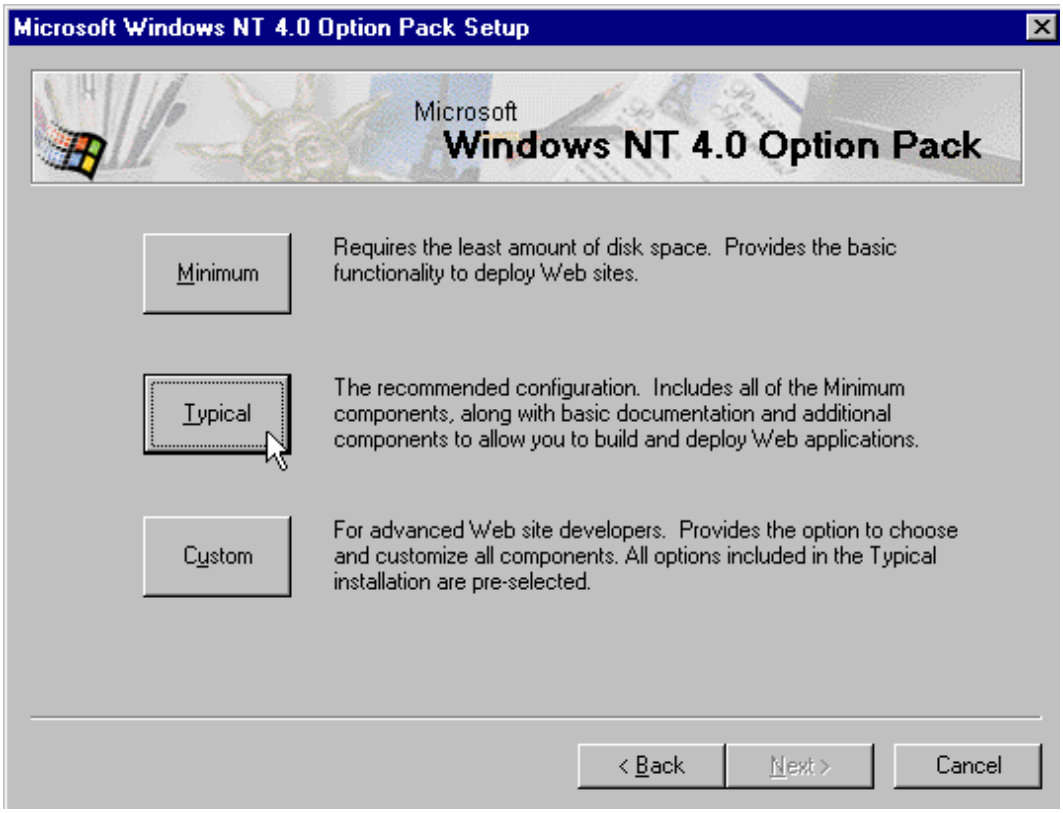

Microsoft Windows NT 4.0 Option Pack Setup
Microsoft
Windows NT 4.0 Option Pack
Minimum
Requires the least amount of disk space. Provides the basic functionality to deploy Web sites.
Typical
The recommended configuration. Includes all of the Minimum components, along with basic documentation and additional components to allow you to build and deploy Web applications.
Custom
For advanced Web site developers. Provides the option to choose and customize all components. All options included in the Typical installation are pre-selected.
< Back
Next >
Cancel

Figure 2.8. Choosing the Components To Be Installed on Your Personal Web Server

Installing ProcessView

Before installing WebHMI, you must have a licensed ProcessViewinstalled on the designated WebHMI server machines.

System Requirements
To use the ProcessViewsoftware, you must have the following *minimum* system requirements:

- 8X Speed CD-ROM
- A minimum of 600 MB disk space available for installing the default ProcessViewsoftware. (Note: Actual amount required may decrease if help and example files are not installed. Project requirements for logging data, alarms and screen storage are above this minimum.)
- VGA Video Card; 256 or more colors for best results.
- Microsoft Internet Explorer 6.0 and above.
- Microsoft Data Access Components (MDAC) 2.8
- Windows 2000, Windows XP, Windows Server 2003, or Windows NT® 4.0 with latest service packs.

Note: Windows 98 and Windows ME are not supported by the Unicode version of ProcessView.

Note: WebHMI Version 8.0 does NOT support Netscape Navigator.
Minimum computer CPU and RAM requirements depend on the application and operating system, as shown in the following table.

Application Size	Processor	RAM
Small	600 MHz	128 MB
Medium	1.2 GHz	256 MB
Large	2.5 GHz	512 MB

The actual amount of RAM and/or processor speed will vary depending upon the I/O counts, networking, logging and alarming requirements, as well as other factors. To determine which type of computer best fits your application, set up a test application station.

Note: The processor and memory requirements of your existing project may be greater in this version of ProcessViewthan in previous versions due to additional and enhanced features in the product.

Note: If your operating system requires a login name, you must log in with administrator capability before installing ProcessView software.

For complete information about installing ProcessView, please see the *Getting Started* Help documentation on the ProcessView product CD.

Installing the ProcessView License Utility

You must install and run the Smar Software Licensing before you can run ProcessView software in Windows in non-demo mode. A Software License Site Key will be generated to handle licensing of ProcessView without requiring a hardware protection key. Licensing encrypting information is stored deep within the Windows systems. See the section below for further details on activating the license. This section first details how to install the software from the CD.

Note: Set the System Date and Time properly before installing the Software License Utility. Setting the time and/or date by more than an hour after the installation will affect any time-limited Licenses (such as the Temporary 30-day Activation).

Note: If your operating system requires a login name, you must log in as "Administrator" before installing the ProcessView Software Licensing.

For complete information about installing Smar Software Licensing, please see the *Getting Started* Help documentation on the ProcessView product CD.

Installing WebHMI

This section describes the steps for installing Smar WebHMI. Be sure to close any other applications before installing WebHMI.

1. To install WebHMI from the ProcessView Product CD, select **WebHMI** from the software installation menu.
2. As you begin installing WebHMI on a properly configured PC, a **Welcome** dialog box appears. Click **Next** to continue the installation.
3. The Smar **WebHMI License Agreement** dialog box will appear, as shown in **Figure 2.9.** Carefully read the Smar License Agreement. If you agree with the terms and conditions, click **Yes** to continue.

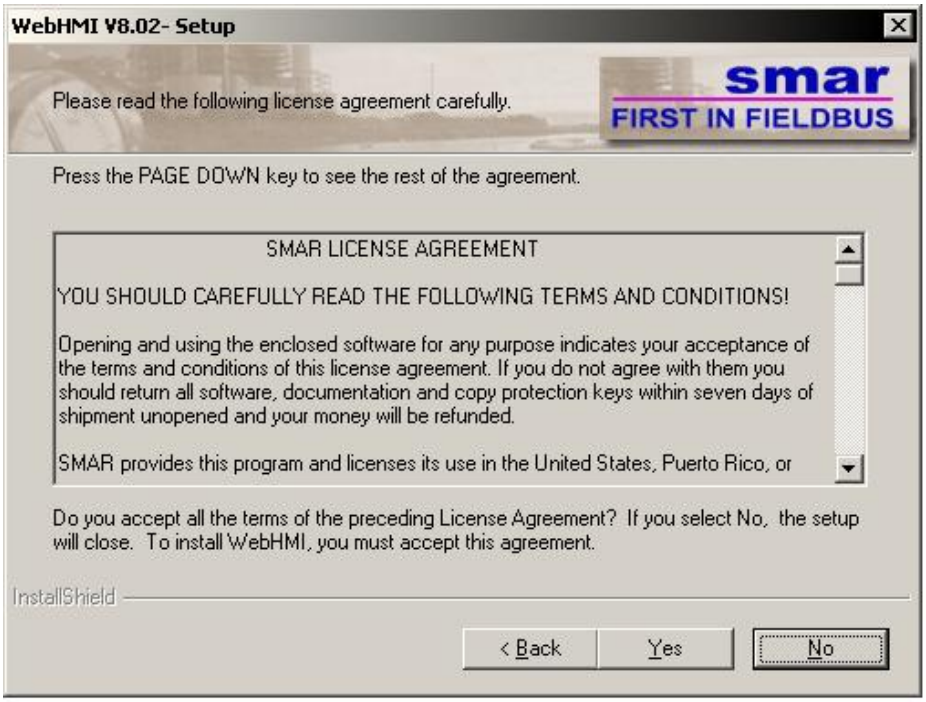

WebHMI V8.02- Setup
Please read the following license agreement carefully.
smar
FIRST IN FIELDBUS
Press the PAGE DOWN key to see the rest of the agreement.
SMAR LICENSE AGREEMENT
YOU SHOULD CAREFULLY READ THE FOLLOWING TERMS AND CONDITIONS!
Opening and using the enclosed software for any purpose indicates your acceptance of the terms and conditions of this license agreement. If you do not agree with them you should return all software, documentation and copy protection keys within seven days of shipment unopened and your money will be refunded.
SMAR provides this program and licenses its use in the United States, Puerto Rico, or
Do you accept all the terms of the preceding License Agreement? If you select No, the setup will close. To install WebHMI, you must accept this agreement.
InstallShield
< Back
Yes
No

Figure 2.9. License Agreement

4. The Smar **WebHMI Copyright** dialog box will appear, as shown in **Figure 2.10.** After reviewing the copyright information, click **Next** to continue.

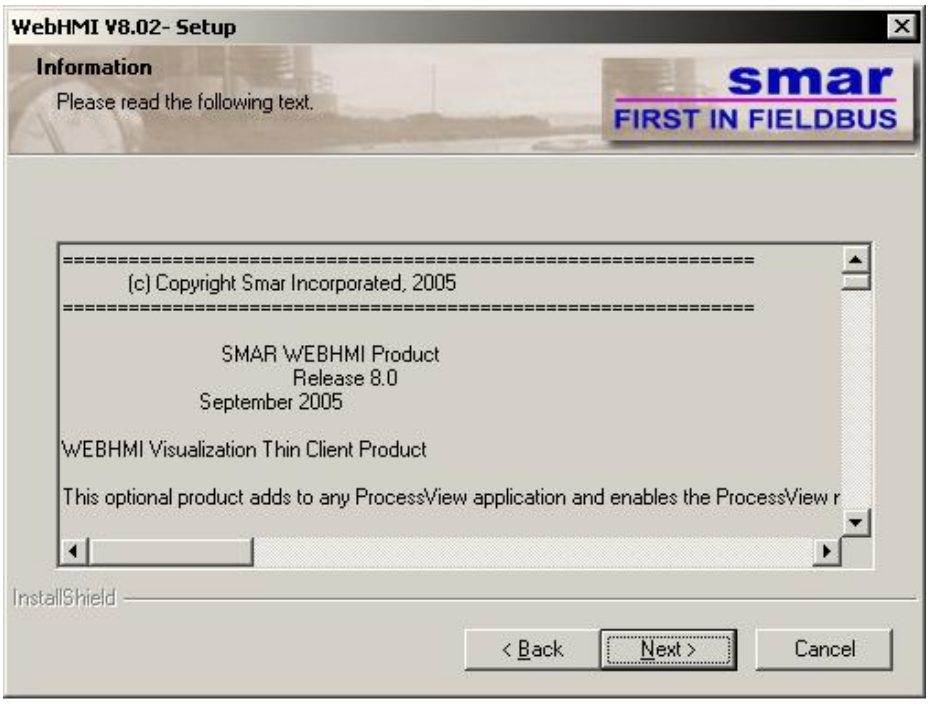


Figure 2.10. Copyright Notice

5. The **User Information** dialog box will appear, as shown in **Figure 2.11.** Type your (the user's) name in the **Name** field and your company's name in the **Company** field.

Figure 2.11. User Information Dialog Box

6. The **Setup Type** dialog box will appear, as shown in **Figure 2.12.** Choose **Typical, Compact,** or **Custom** installation. **Typical** installs all WebHMI components. **Compact** installs no sample pages. **Custom** lets you select the specific components to be installed. After making your choice, click the **Next** button to continue.

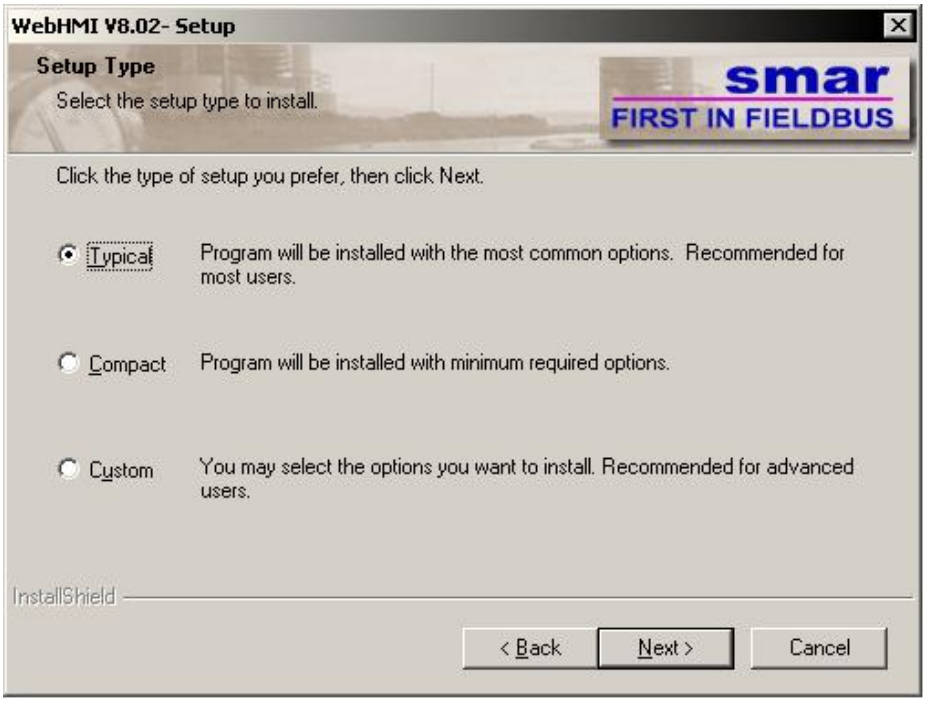

WebHMI V8.02- Setup
Setup Type
Select the setup type to install.
smar
FIRST IN FIELDBUS
Click the type of setup you prefer, then click Next.
Typical
Program will be installed with the most common options. Recommended for most users.
Compact
Program will be installed with minimum required options.
Custom
You may select the options you want to install. Recommended for advanced users.
InstallShield
< Back
Next >
Cancel

Figure 2.12. Choosing the Installation Type

7. The **Choose Communication Type** dialog box will appear, as shown in **Figure 2.13.** Choose **OPC over TCP/IP, OPC over DCOM,** or **OPC over SOAP/XML** for installation.

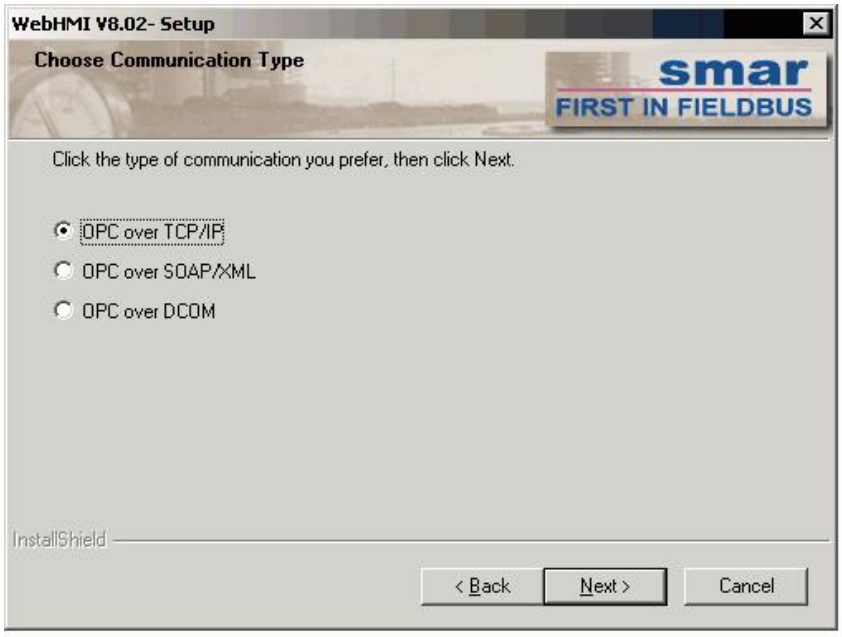


Figure 2.13. Choosing a Communication Type

Since the default communication type is **OPC over TCP/IP,** it is recommended that you use it to install WebHMI server. If desired, you will then be able to change the settings at any time. **OPC over TCP/IP** is easy to use on a local area network (LAN), over an intranet or the Internet, and communicates through routers and firewalls. Both **SOAP/XML** and **TCP/IP** communicate over intranets/Internet through routers and firewalls. For information about configuring TCP/IP and SOAP/XML channels, please see the GenBroker Help documentation.

OPC over DCOM is recommended for use on a local area network where you know that all nodes have been properly set up. It is also recommended for use if you wish to avoid connecting over the Internet because you are uncertain of how all nodes in a network have been set up.

Note: OPC over DCOM works only in an environment where Microsoft Network has been installed. If you wish to communicate over the Internet, install OPC over TCP/IP.

After you have chosen the communication type that you prefer, click **Next.** The **Select Program Folder** dialog box will appear, as shown in **Figure 2.14.**

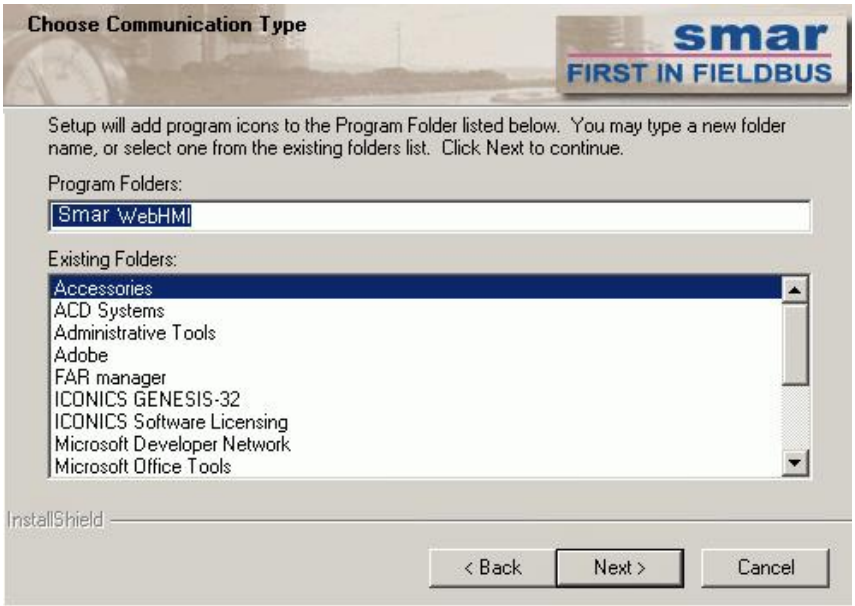


Figure 2.14. Select Program Folder

Either type the name of a new program folder in the **Program Folders** field or select the program folder of your choice from the list in the pane below. Click the **Next** button.

Note: If you wish to change your choice of communication types, it is recommended that you uninstall WebHMI and perform the installation again from the beginning.

8. The **Copying System Files** window will appear. While the system files are being automatically copied, the percentage of the files already copied will appear in the window's progress bar.

9. After all of the necessary system files have been copied, a dialog box will appear, as shown in **Figure 2.15.** Enter the name of your WebHMI server or select a local machine name. Customize your application by selecting one of three choices under **Name Convention:**

- **Microsoft Network.** This is the name of the WebHMI server machine located on your local network. This is recommended with DHCP.
- **DNS (Domain Network Service).** This is your registered domain name, such as www.myserver.com.
- **IP Address.** The numeric IP address, such as 99.99.99.99 will automatically appear in the **DNS Name** field.

After you have customized your application, click the **OK** button.

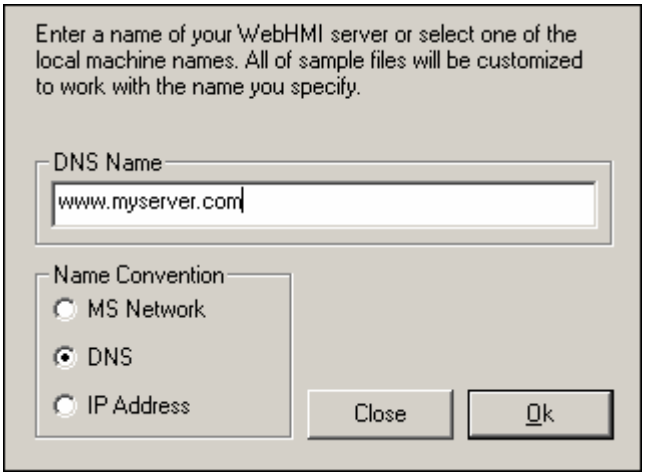

Enter a name of your WebHMI server or select one of the local machine names. All of sample files will be customized to work with the name you specify.
DNS Name
www.myserver.com
Name Convention
MS Network
DNS
IP Address
Close
Ok

Figure 2.15. Specifying the WebHMI Server Host Name

10. The **Setup Complete** dialog box will appear. Click the **Finish** button to complete the setup.

You have successfully completed installation of WebHMI on your computer.
.

Chapter 3

VIEWING WEBHMI SAMPLES

Introduction to WebHMI Samples

The WebHMI installation includes a series of sample screens, which get installed in the *Inetpub\wwwroot\WebHMI\Samples* directory. To access the Samples, select **Start > Programs > Smar WebHMI > WebHMI Examples.** The **WebHMI Samples Home Page** will appear, as shown in **Figure 3.1.** This page allows you to access both default (basic) and advanced WebHMI samples, as well as industry examples.

Note: When navigating the displays, you can click on the House icon at the top right of the screen to return to the home page at any time.

smar
WebHMI
Smar' WebHMI enables users to view real-time HMI or SCADA applications over the Internet or intranets on any client PC running Microsoft Internet Explorer or Netscape Navigator. WebHMI is extremely easy to use and set up.
Advanced Web publishing tools allow graphics, trends, alarms and reports to be published to designated Web servers running Microsoft Internet Information Server (IIS), Microsoft Personal Web Server, or Apache Server providing a complete Web-based HMI or SCADA solution.
"Our Web-enabled HMI and SCADA products are based on OPC-To-The-Core technology and offer unparalleled performance, efficiency, ease of connectivity and freedom of choice!"
Default Samples
Industry Samples
Advanced Samples
text inputbar
change color
Hide rectangle
read value
English

Figure 3.1. WebHMI Samples Home Page

Default Samples

Clicking **Default Samples** on the WebHMI Samples Home page opens the WebHMI **Default Samples** page, shown in **Figure 3.2.**

Click on the back arrow button at the bottom left at any time to move back to the previous display.

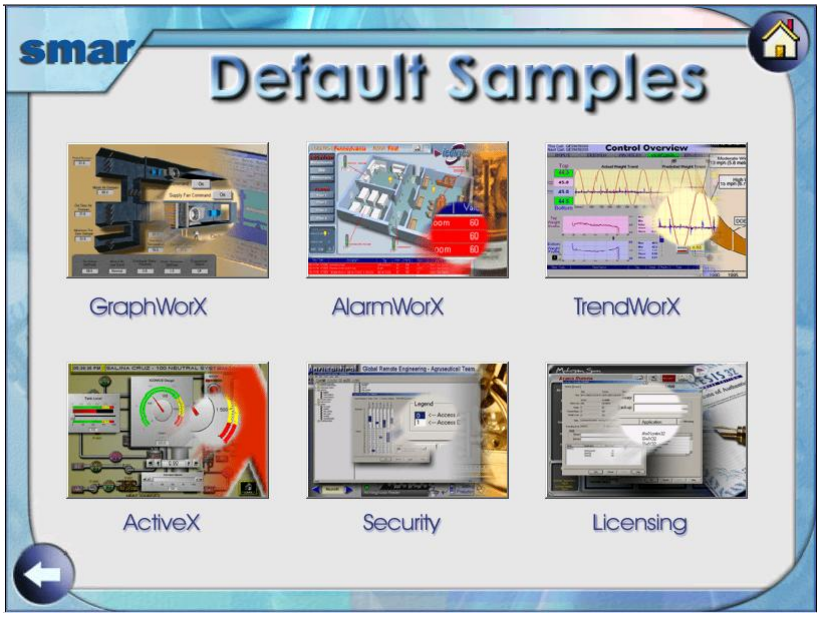


Figure 3.2. WebHMI Default Samples Page

GraphWorX Samples

Click on the **GraphWorX** link on the left-hand side of the home page. The **GraphWorX** sample page will appear, as shown in **Figure 3.3.**

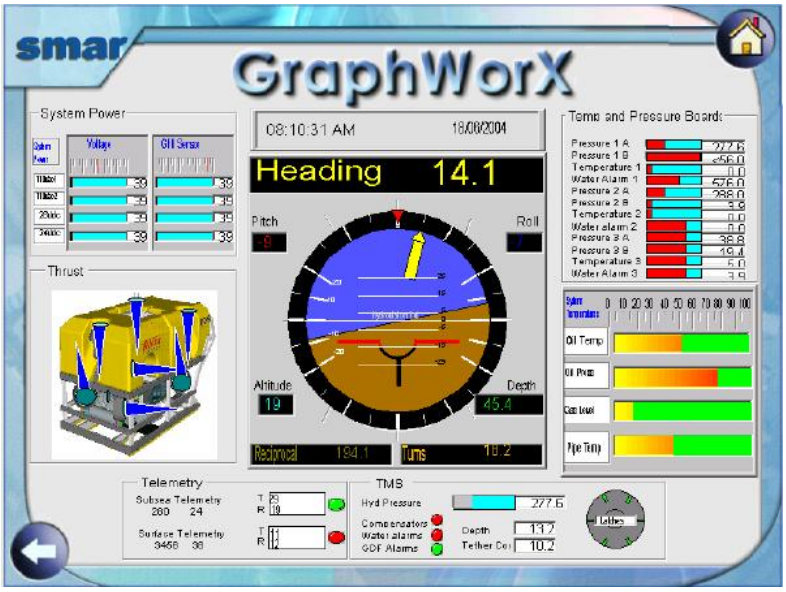


Figure 3.3. GraphWorX Sample Page

AlarmWorX Samples

Return to the Default Samples page and click on the **AlarmWorX** link. The **AlarmWorX** sample page will appear, as shown in **Figure 3.4.** This contains example screens for the Alarm Viewer ActiveX, the Alarm Report ActiveX, and an Alarm Chart. You can move the mouse pointer over the alarms in the display to show ToolTips for the alarms, which provide information about each alarm.

Note: You must have MSDE installed to view the Alarm Report and Alarm Chart samples.

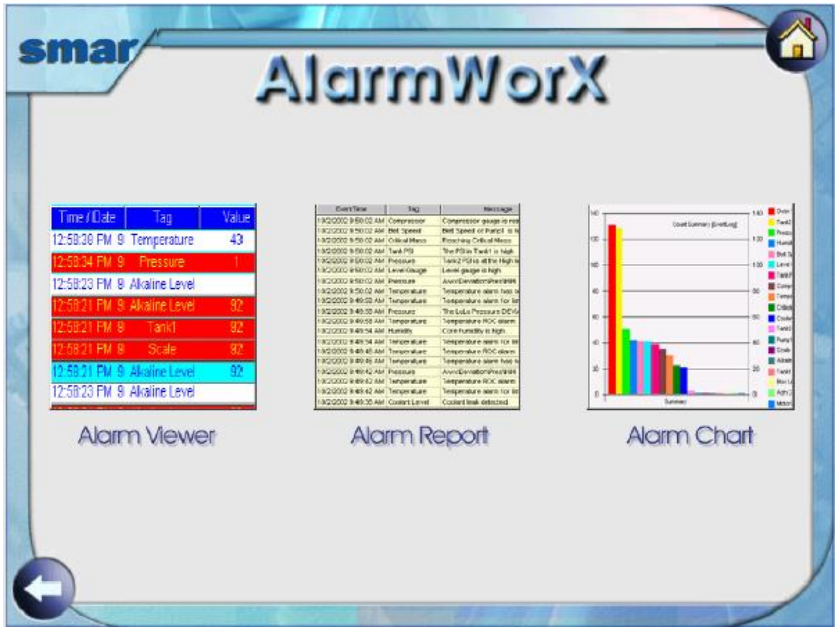


Figure 3.4. AlarmWorX Samples Page

TrendWorX Samples

Return to the Default Samples page and click on the **TrendWorX** button. The **TrendWorX** sample page will appear, as shown in **Figure 3.5.** Here you can view various trend displays with trend pens in runtime mode, including a Time Plot, a Log Plot, a Strip Chart, a Circular Chart, and an XY plot.

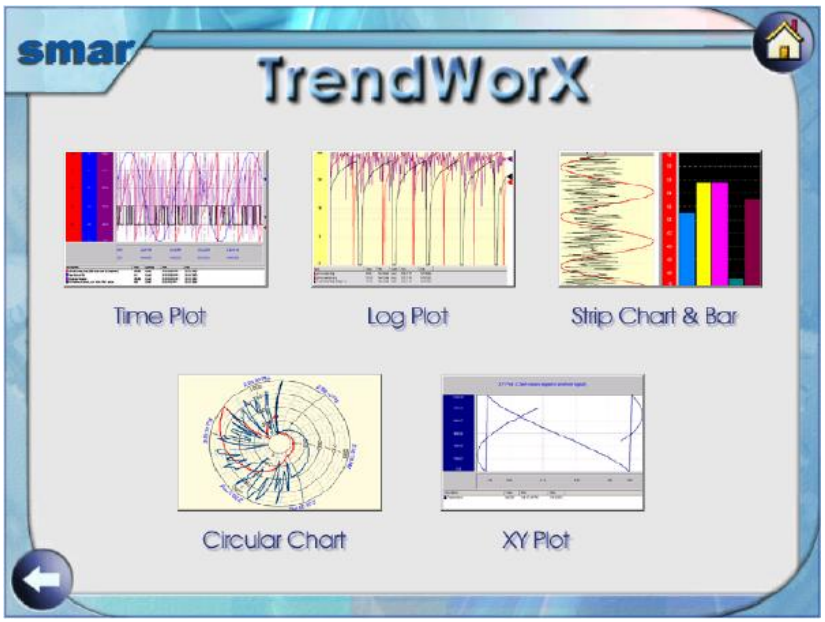


Figure 3.5. TrendWorX Sample Pages

ActiveX Control Samples

Return to the Default Samples page and click on the **ActiveX** button. The **ActiveX** sample page will appear, as shown in **Figure 3.6.** To view various types of Smar ActiveX controls in action, including the Switch, Slider, Vessel, Gauge, and Numeric ActiveX controls, click on the buttons.

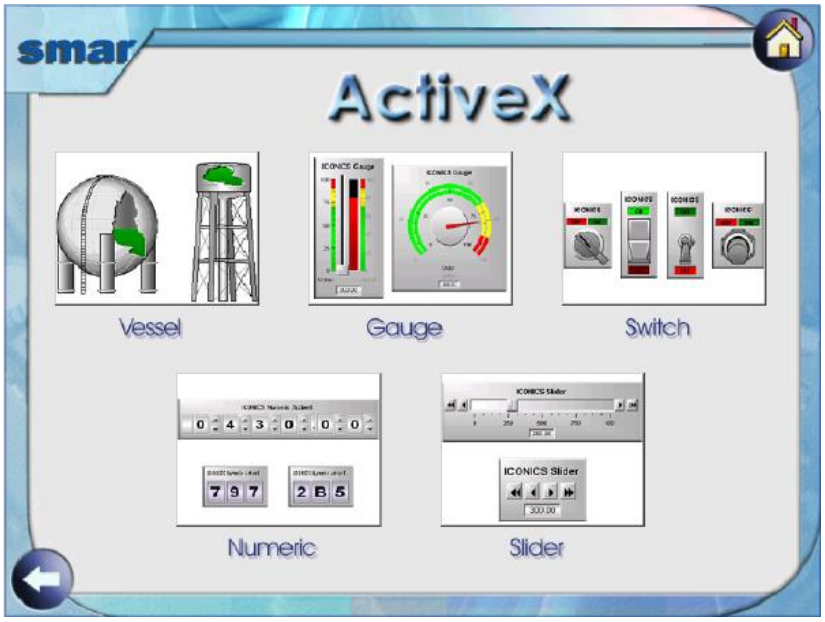


Figure 3.6. ActiveX Controls Samples

Licensing and Security Samples

The WebHMI Default Samples page also contains links to the **Licensing** and **Security** sample pages. For information on Security and Licensing, please see Chapter 6 and Chapter 7, respectively.

Advanced Samples

Return to the WebHMI Samples Home page and click on the **Advanced Samples** button, as shown in **Figure 3.7.**

smar
WebHMI
Smar WebHMI enables users to view real-time HMI or SCADA applications over the Internet or intranets on any client PC running Microsoft Internet Explorer or Netscape Navigator. WebHMI is extremely easy to use and set up. Advanced Web publishing tools allow graphics, trends, alarms and reports to be published to designated Web servers running Microsoft Internet Information Server (IIS), Microsoft Personal Web Server, or Apache Server providing a complete Web-based HMI or SCADA solution.
"Our Web-enabled HMI and SCADA products are based on OPC-To-The-Core technology and offer unparalleled performance, efficiency, ease of connectivity and freedom of choice!"
Default Samples
Industry Samples
Advanced Samples

Figure 3.7. WebHMI Samples Home Page

This opens the Advanced Samples page, shown in Figure 3.8. The Advanced Samples page contains two buttons: Remote Connection and Advanced Scripting.

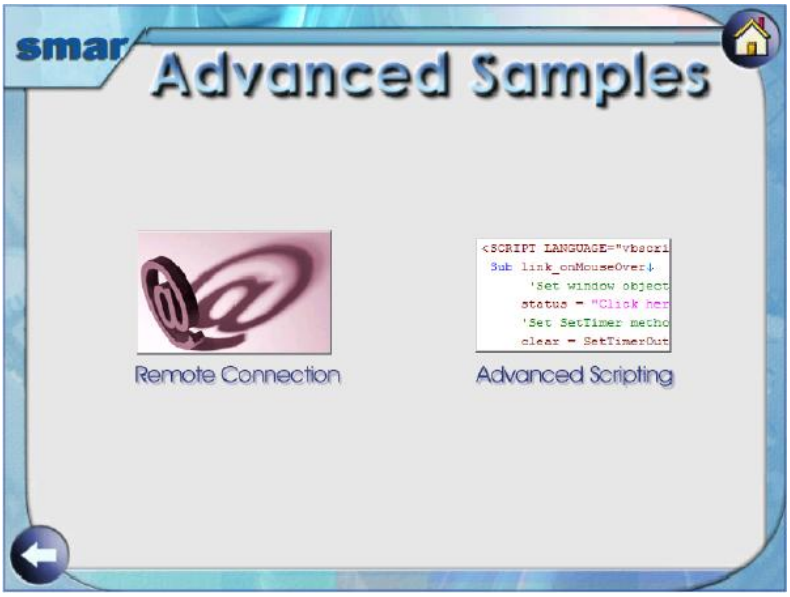


Figure 3.8. Advanced Samples Page

The **Remote Connection** samples page, shown in **Figure 3.9,** contains examples of complex GraphWorX display (.gdf) files that have been published as HTML pages through WebHMI. You can view these display files in real time over your WebHMI server. The examples include a bar graph, an Alarm Viewer, and a Trend Viewer.

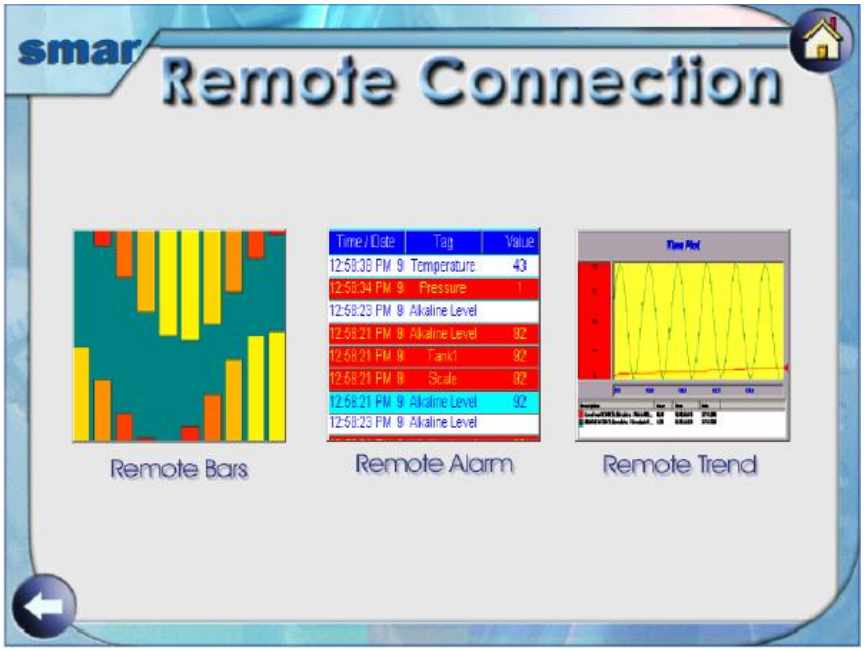


Figure 3.9. Remote Connection Page

The **Advanced Scripting** samples page, shown in **Figure 3.10,** contains examples of the use of VBScript in GraphWorX displays. For more information about the GraphWorX scripting engine, please see the GraphWorX Help documentation.

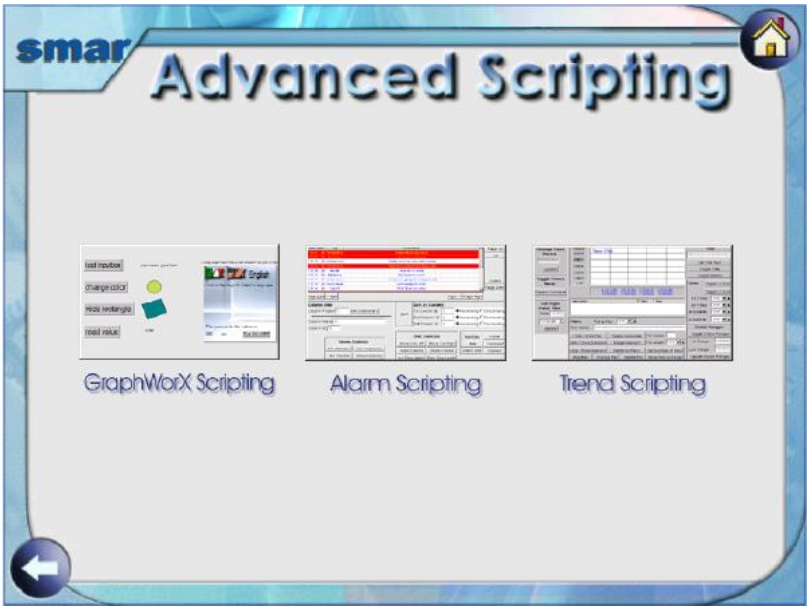


Figure 3.10. Advanced Scripting Page

Industry Samples

Return to the WebHMI Samples Home page and click on the **Industry Samples** button. This opens the Industry **Samples page**, shown in **Figure 3.11,** which contains example GraphWorX displays for various segments of the automation industry.

Figure 3.11. Industry Samples Page

Chapter 4

WEB PUBLISHING WIZARD

Introduction to HTML Publishing

The ProcessView **Web Publishing Wizard** enables you to "export" your GraphWorX (.gdf), TrendWorX (.t32), and AlarmWorX (.a32) displays to HTML files and/or publish the HTML files to a Web server (LAN or Internet). In publishing displays to a Web server, WebHMI uses HTML to reference the files in an Internet-enabled format. Once a display is "exported" to an HTML file and then published to a Web server, client machines can browse it through an Internet browser, such as Microsoft Internet Explorer. Each display can be viewed as a Web page.

Basic Steps in Publishing GraphWorX Displays

Publishing a GraphWorX display to HTML involves the following basic steps:

1. Create a GraphWorX display (.gdf) file on a developer workstation with GraphWorX installed.

2. Use the Web Publishing Wizard to generate an HTML file and publish the file to a Web server (URL) address.

3. On the Web server PC (with ProcessView installed), start GenBroker Server from ProcessView Tray.

4. On a client PC, open a Web browser, such as Microsoft Internet Explorer, and browse to the URL address of the HTML file. The client PC downloads the HTML file from the Web server. The Web server delivers all the necessary components (e.g. ActiveX controls) to the client PC's Web browser, which then runs the GraphWorX display directly within the browser. This way you can view the GraphWorX display in real time as a Web page.

The source code of the published HTML file contains references to the required “plug-ins” that are needed to deliver the GraphWorX Viewer ActiveX control to the client. The GraphWorX Viewer ActiveX is needed to run the .gdf file over the network. The GraphWorX display itself is not really “converted” into HTML. Instead the display (.gdf) file is referenced in the HTML code by the ActiveX plug-in.

To expand on step 2 above, the following Microsoft Internet Explorer example shows the portion of the generated HTML source code that references the .gdf file. The **GWXview32.cab** file is referenced in the **codeBase** field to deliver the “plug-in” for the GraphWorX Viewer ActiveX. This in turn opens up the .gdf display. In the example below, the .gdf display file is called **Building1.gdf,** as shown below in the **DisplayName** parameter field.

```
<OBJECT classid=clsid:98A5DDE3-563B-11CF-A343-487C03C10000
codeBase="http://www.myserver.com/webhmi/cabs/GWXview32.cab"
id=GWXview321 style="HEIGHT: 420px; WIDTH: 620px" height=420 width=620>
<param name="_Version" value="65537">
<param name="_ExtentX" value="16404">
<param name="_ExtentY" value="11113">
<param name="_StockProps" value="160">
<param name="BorderStyle" value="1">
<param name="Appearance" value="1">
<param name="DisplayName" value="http://www.myserver.com/webhmi/Building1.gdf">
<param name="UseAmbientBackColor" value="0">
<param name="AutoStartRuntime" value="-1">
<param name="OverrideScrollbarSettings" value="1">
<param name="VerticalScrollbar" value="0">
<param name="HorizontalScrollbar" value="0">
</OBJECT>
```

Exporting a .gdf file to an HTML file not only references the GWXview32.cab file, as shown in the example above, but also references any other "plug-in" .cab files required for other ActiveX components that may be embedded within the .gdf file (e.g., TrendWorX or AlarmWorX Viewer ActiveX controls).

However, before a .gdf file can communicate with live OPC data, the **IcoSetServer.cab** file "plug-in," which is necessary for security and licensing, as well as a **GenBroker configuration (.gbx) file,** which contains network configuration settings for OPC communications, must also be delivered to the client PC. The IcoSetServer.cab file and the .gbx file are referenced in the HTML source code. This way the security and licensing information are available whenever a component (such as an Alarm Viewer ActiveX) is downloaded. The sample HTML source code below shows how these files are referenced in the code.

```
<object id="SetServer" classid="clsid:57802C16-9A15-11D4-B2A8-0090272E599B"
codeBase=http://www.myserver.com/WebHMI/cabs/IcoSetServer.cab height=28 width=17>
   <PARAM NAME="CfgName"
VALUE="http://www.myserver.com/WebHMI/Samples/Default.gbx">
</object>
```

The ProcessView Web Publishing Wizard in GraphWorX takes care of all the necessary HTML code references automatically. Microsoft Internet Explorer uses .cab files.

Note: If you are using multiple frames for your WebHMI pages (e.g. a main "navigation" frame for browsing between pages, and a "content" frame that contains the body of the pages), make sure that the IcoSetServer.cab file and the .gbx file are properly referenced in the HTML source code for the main frame. This way the security and licensing information are available whenever a component is downloaded.

Delivering the Necessary Web Components to the Client

WebHMI is designed to operate with **Zero Install** and **Thin Client** philosophy. This means the client PC has nothing but Windows and Internet Explorer loaded. Thus, all necessary Web components must be remotely delivered and seamlessly installed. The type and number of components required for delivery on the client PC are determined by the content of the GraphWorX displays. For instance, a display with an embedded Trend Viewer requires delivery of the corresponding TrendWorX Viewer ActiveX Web component, but not the AlarmWorX Viewer ActiveX. On the other hand, a display with an embedded Alarm Viewer requires the corresponding AlarmWorX Viewer ActiveX but not the TrendWorX Viewer ActiveX.

The delivery process can be very slow in terms of network speed. To avoid wasting time, the first requirement is to deliver only the required component. The delivery process is strictly connected to the HTML code. Inside the HTML code, there are references to .cab files, the basic ways to deliver components. As a result, the building of the HTML page is a critical step. The Web Publishing Wizard looks inside the display to understand what components are needed on the client side and then builds the corresponding HTML code.

Client Station Requirement
In order to view a published GraphWorX display, a client PC must have Microsoft Internet Explorer installed.

Developer Station Requirement
The developer station requires GraphWorX.

Web Server Station Requirement
The Web server PC must have the following installed:

- WebHMI
- ProcessView (with GenBroker)
- For Windows NT, Internet Information Server (IIS) or a Personal Web Server
- For Windows 2000, Windows XP and Windows Server 2003, Internet Information Server (IIS)

Note: Because publishing via Microsoft Posting Acceptor is no longer supported on Windows Server 2003, we were forced to find another way of transporting published files from a ProcessView development machine to a WebHMI server. It appeared that FTP would be the best choice. For more information, please see the "Publishing Channel" section below.

Multiple Display Support

Suppose that you have a main GraphWorX display (.gdf) file that is linked to other display files (e.g. each display contains pick actions, such as Load Display or Pop-up Window, that point to the other display files). When you publish the main display to an HTML file, you want all the links and references to the other dependent display files to be functional when the HTML file is downloaded to a client Web browser. The Web Publishing Wizard makes this possible by detecting all mutually

linked display files, looping through all dynamic actions, and checking for pick actions in which a file name is specified as one of its parameters. The following pick actions are supported for multiple GraphWorX display Web publishing support:

- Load Display
- Embedded Window
- Popup Window
- Drag/Drop Load
- Set Aliases
- Alias Dialog (alias files specified through the **Set Aliases** dialog box in GraphWorX).

The LoadTabs display property is also supported for publishing multiple GraphWorX displays.

Embedded ActiveX Control Support

In dealing with a main GraphWorX display (.gdf) file that is linked to other display files (as described in the "Multiple Display Support" section above), suppose you included an ActiveX Control (e.g. Alarm Viewer ActiveX or Trend Viewer ActiveX) into each of the dependent display files. Also suppose that you generated and published an HTML file only from the main display, but you want this single HTML file to trigger the simultaneous download and installation of all ActiveX Controls embedded within all interlinked GraphWorX display files. The Web Publishing Wizard's embedded ActiveX Control support makes this possible, allowing you to view the ActiveX Controls in all displays from a single client Web browser.

Using the Web Publishing Wizard

The Web Publishing Wizard performs two basic operations:

1. The Wizard creates an HTML file based on a user-specified GraphWorX display (.gdf) file, TrendWorX display (.t32) file, or AlarmWorX display (.a32 file).

2. The Wizard then either "exports" (saves) the HTML file to a user-specified directory on the local drive and/or "publishes" (uploads) the HTML file to a user-specified Web server URL address (i.e. over the Internet or an intranet).

Starting the Web Publishing Wizard

To launch the Web Publishing Wizard in GraphWorX:

1. Load or create a GraphWorX display (.gdf) file.

2. After you have either created a new display file or opened an existing file, select **Save As** from the **File** menu. The **Save As** dialog box opens, allowing you to specify the name and location of the file you are about to save. Type a name for the new file in the **File Name** field. It is important that, when saving your file, you select "GraphWorX Displays without VBA (*.gdf)" in the **Save as Type** field, as shown in **Figure 4.1,** because the Web Publishing Wizard does not support Microsoft Visual Basic for Applications (VBA). Click the **Save** button to save the current file.

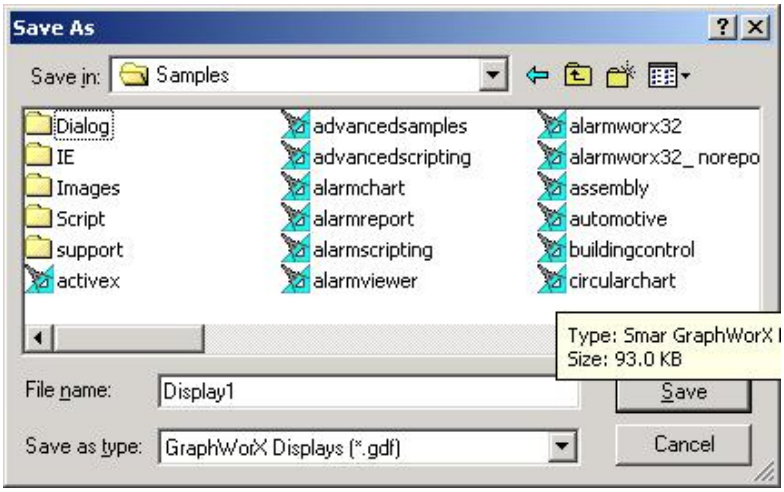


Figure 4.1. Saving the Display As a Non-VBA File

3. Select **Publish to HTML** from the **Tools** menu, as shown in **Figure 4.2.**

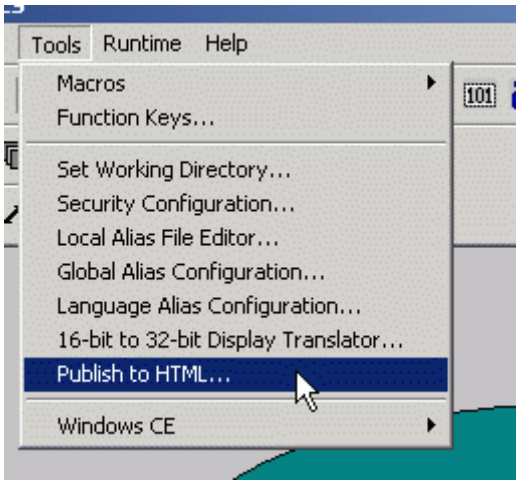


Figure 4.2. Starting the Web Publishing Wizard in GraphWorX

Note: If you try to run the Web Publishing Wizard with a GraphWorX display with VBA, you will get a warning message as shown in **Figure 4.3.** If you click **Yes,** the Web Publishing Wizard automatically saves the display as a non-VBA file. If you click **No,** the publishing operation is cancelled.

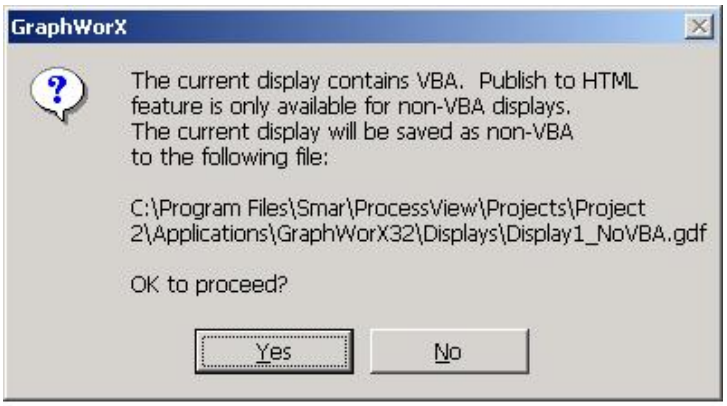

GraphWorX
The current display contains VBA. Publish to HTML feature is only available for non-VBA displays. The current display will be saved as non-VBA to the following file:
C:\Program Files\Smar\ProcessView\Projects\Project 2\Applications\GraphWorX32\Displays\Display1_NoVBA.gdf
OK to proceed?
Yes
No

Figure 4.3. Warning Message About Displays With VBA

4. This launches the **Web Publishing Wizard** configuration dialog box, as shown in **Figure 4.4.** This dialog box serves as the interface through which you export/publish GraphWorX display files to the Web.

Note: Both the **Publish to Web Server** and **Export Local Copy** actions can be performed at the same time.

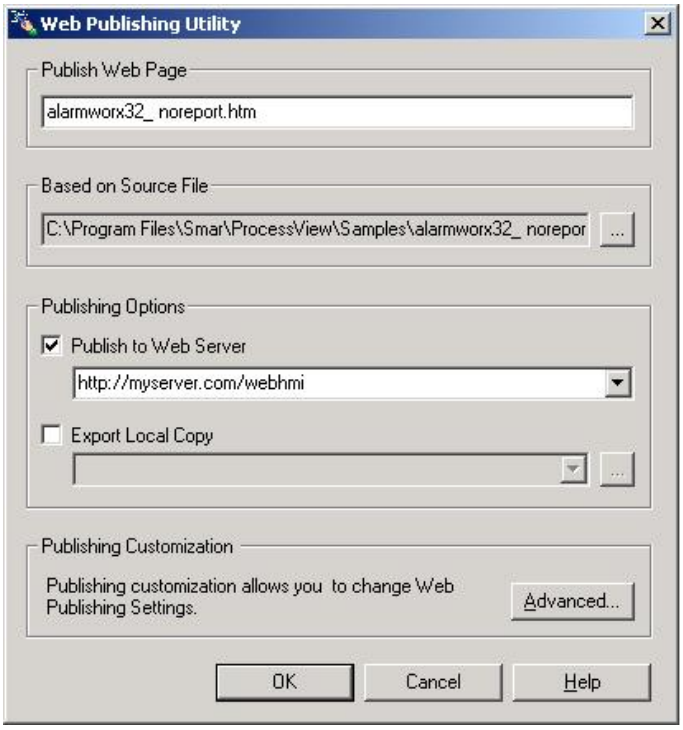


Figure 4.4. Web Publishing Wizard Dialog Box

Exporting a Display File Locally

To export a GraphWorX display to a directory on the local drive:

1. In the **Based on Source File** field of the Web Publishing Wizard, you must specify the name of the GraphWorX display (.gdf) file to be exported, as shown in **Figure 4.5.** To choose a different display, click the **…** button to the right to browse for a file. Select a file and then click **Open.** The directory path and the file name are shown in the text field.

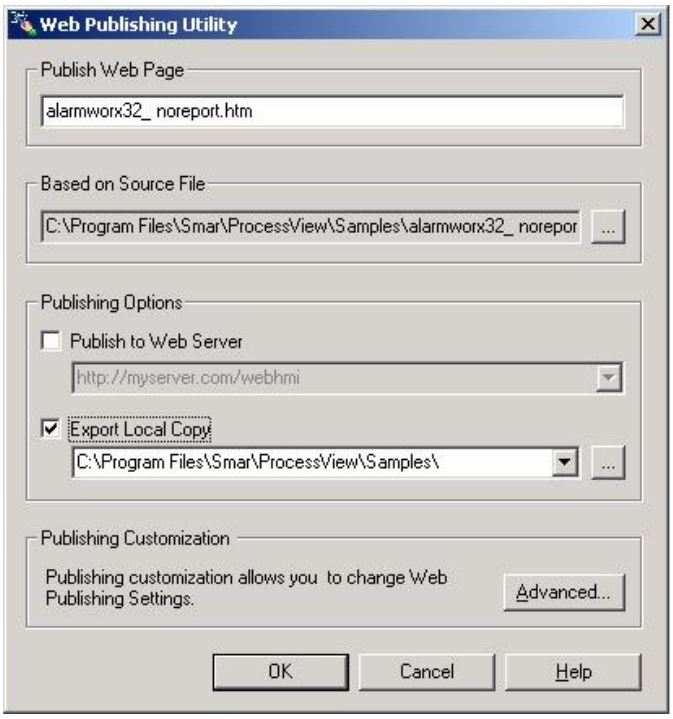


Figure 4.5. Specifying a GraphWorX Display File

2. In the **Publish Web Page** field of the Web Publishing Wizard, specify the name of the HTML file that will be created. The .gdf file name is filled in by default, but you can give the HTML file a different name.

3. In the **Publishing Options** field, check the **Export Local Copy** check box and specify the local directory path name to which you want to export the HTML file. You can select a recently used path from the drop-down list, or click the **…** button to browse for a destination directory, as shown in **Figure 4.6.** Select the directory and click **OK.**

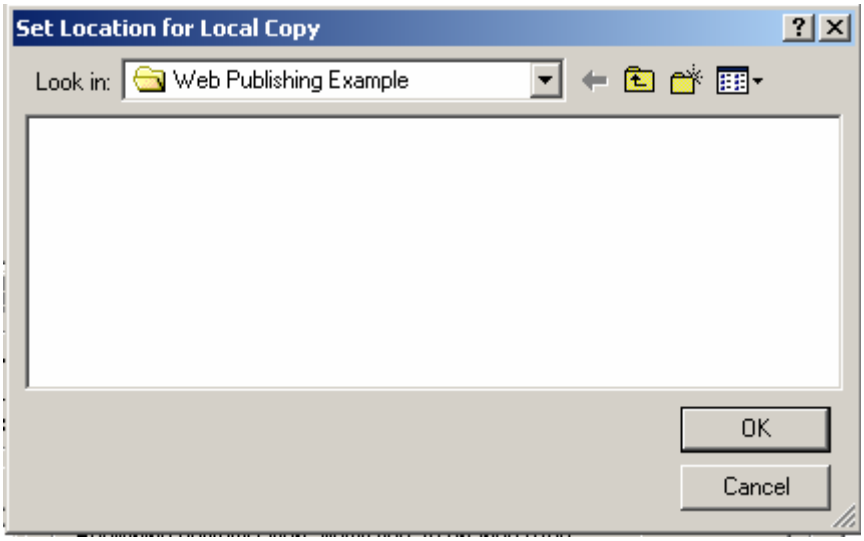


Figure 4.6. Setting the Local Directory

4. The local directory pathway you selected appears in the **Export Local Copy** field of the Web Publishing Wizard, as shown in **Figure 4.7.**

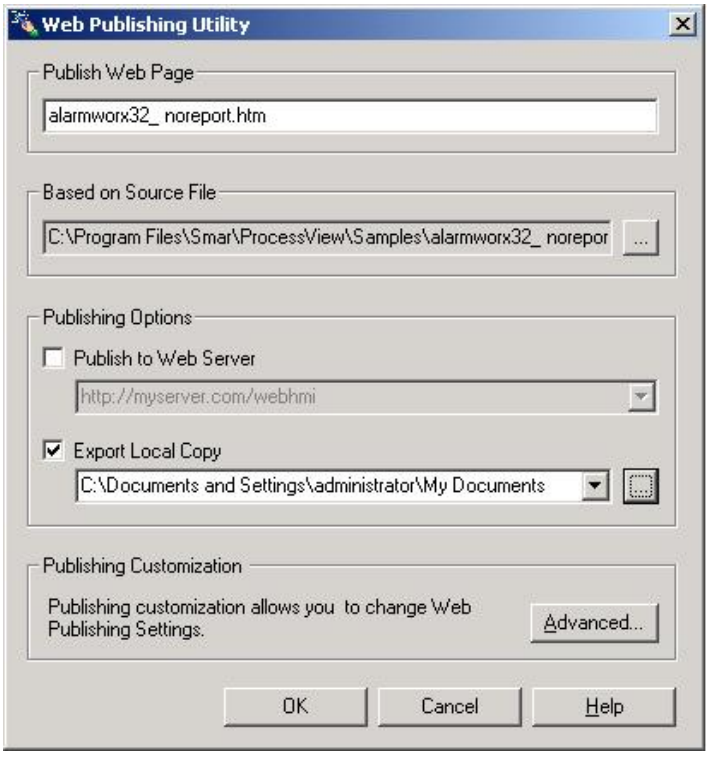


Figure 4.7. Export Local Copy Pathway Shown in the Wizard

5. Click the **OK** button to generate the HTML file. The new HTML file is saved to the local directory pathway specified in the **Export Local Copy** field.

Note: Both the **Publish to Web Server** and **Export Local Copy** actions can be performed at the same time.

Publishing a Display File to a Web Server

To publish the HTML file to a directory on a Web server (i.e. over the Internet or an intranet):

1. In the **Based on Source File** field of the Web Publishing Wizard, you must specify the name of the GraphWorX display (.gdf) file to be published, as shown in **Figure 4.8.** To choose a different display, click the **…** button to the right to browse for a file. Select a file and then click **Open.** The directory path and the file name are shown in the text field.

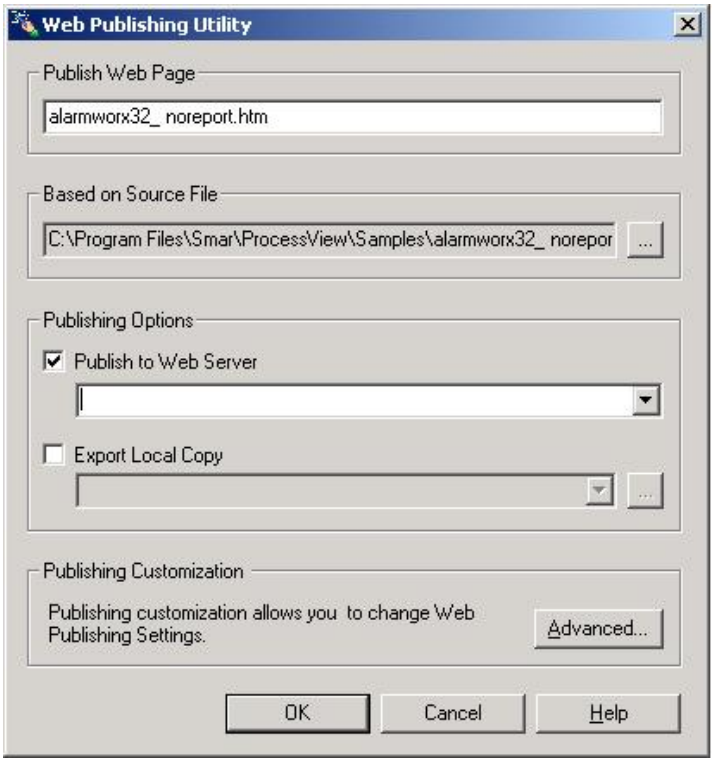


Figure 4.8. Specifying a GraphWorX Display File

2. In the **Publish Web Page** field of the Web Publishing Wizard, specify the name of the HTML file that will be created. The .gdf file name is filled in by default, but you can give the HTML file a different name.

3. In the **Publishing Options** field, check the **Publish to Web Server** check box and type in the URL address of your Web server with the complete directory indicating where you want to publish the HTML file, as shown in **Figure 4.9.** In the sample Web server URL address shown below ("http://www.myserver.com/WebHMI/Samples"), the various components are:

- **http://www.myserver.com:** IP address (server name) of the WebHMI Server
- **WebHMI:** Name of the WebHMI server root directory
- **Samples:** Name of the directory on the server to which the HTML file will be saved

4. The Web server URL address now appears in the Web Publishing Wizard dialog box in the **Publish to Web Server** field of the Web Publishing Wizard, as shown in Figure 4.9.

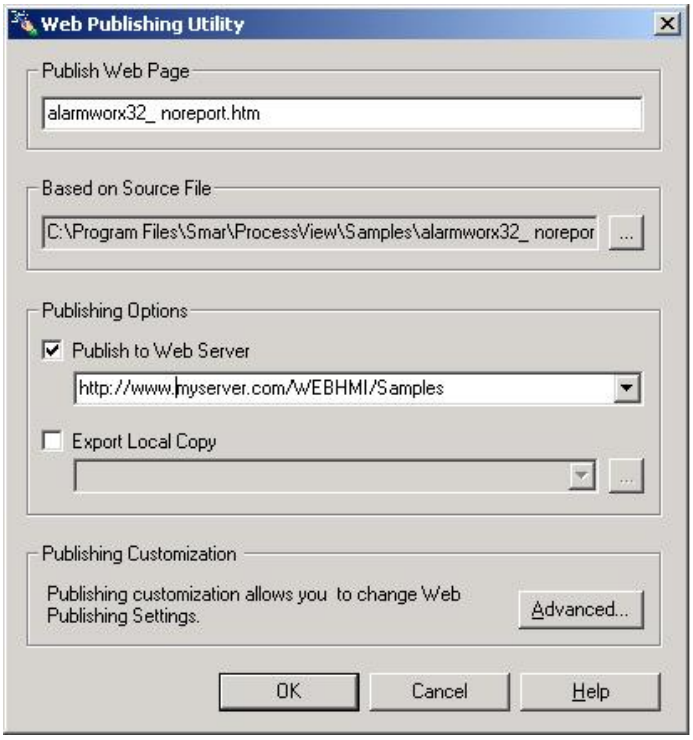


Figure 4.9. Publish to Web Server Location Specified in Web Publishing Wizard

5. Click the **OK** button to generate the HTML file. The new HTML file is uploaded to the Web server and then saved to the URL address path specified in the **Publish to Web Server** field.

Note: Both the **Publish to Web Server** and **Export Local Copy** actions can be performed at the same time.

Publishing Customization Options

The Web Publishing Wizard contains some customization options for exporting/publishing HTML files. Clicking the **Advanced** button on the Web Publishing Wizard dialog opens the **Web Publishing Properties** dialog box, as shown in **Figure 4.10,** which contains the following tabs:

- Server Switching Support
- Screen Resolution Settings
- Publishing Options

General Settings

The **General** tab of the **Web Publishing Properties** dialog box, shown in Figure 4.10, allows you to specify the default URL address for the WebHMI Server. The **General** tab is visible only when **Export Local Copy** is the sole option selected on the Web Publishing Wizard dialog box. If **Publish to Web Server** is selected on the Web Publishing Wizard dialog box, then the **General** tab is hidden and you won't need to configure it.

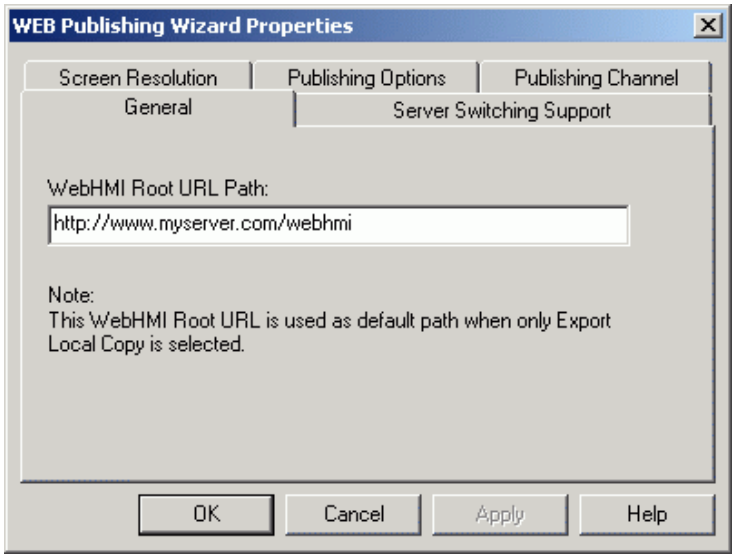


Figure 4.10. General Settings

Server Switching Support

The **Server Switching Support** tab of the **Web Publishing Properties** dialog box, shown in **Figure 4.11,** allows you to enable or disable GenBroker support for the exported/published HTML file. Here you can specify which GenBroker configuration file (.gbx) will be activated. The GenBroker configuration file establishes the settings for OPC data communications between the clients and the Web server.

Figure 4.11.Server Switching Support Settings

There are three available options for GenBroker support:

- **None:** GenBroker support is disabled. If this option is selected, all the communication will default to the local node. This option should be checked when the Web pages are meant to be used only from the local node. If this option is checked, Web pages may not work on a remote node, unless the remote node has ProcessView installed and GenBroker is properly configured to communicate with the WebHMI server.

- **As Configured in WebHMI Settings:** This setting uses the default URL address of the GenBroker configuration (.gbx) file as specified in the WebHMI installation. This is the recommended option for non-expert users.
- **User Defined:** When this option is selected, the **GenBroker Configuration File URL** field becomes available, allowing you to specify the URL address of the GenBroker configuration (.gbx) file. This option gives the maximum flexibility and should be used by expert users only.

Note: For information about configuring GenBroker settings, please see the GenBroker Configurator Help documentation.

Screen Resolution Settings

The **Screen Resolution** tab of the **Web Publishing Properties** dialog box, shown in **Figure 4.12,** determines the screen resolution and size of the GraphWorX Viewer ActiveX control, which is referenced in the generated HTML file and then downloaded to a client PC when the HTML file is viewed in the client's Web browser. You can specify the screen size in the **Width** and **Height** fields in terms of pixels (px) or percentage (%). Click the **Resolutions** button to select from a pop-up menu of standard screen resolutions, as shown in Figure 4.12.

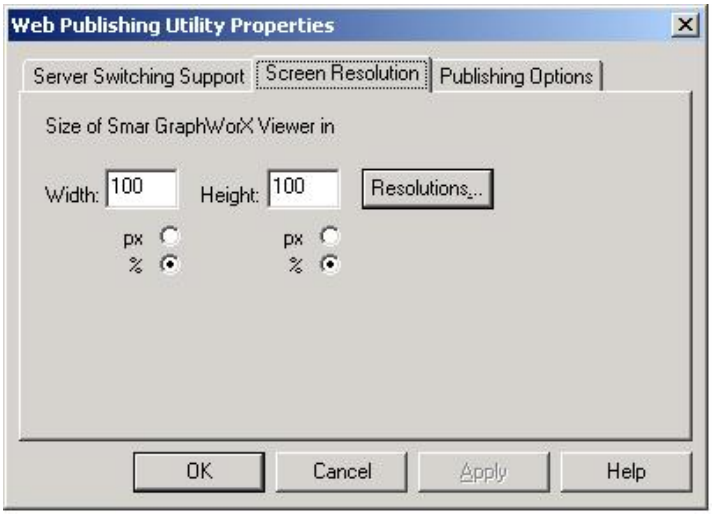


Figure 4.12. Screen Resolution Settings

Publishing Options

The **Publishing Options** tab of the **Web Publishing Properties** dialog box, shown in **Figure 4.13,** enhances the publishing process. By default, the Web Publishing Wizard provides support for related files detection (for more details, see the **Multiple Display Support** section). The multiple display detection could be a rather lengthy process. You can speed up the process by disabling the **Enable Multiple Display Support** option. This is especially useful, if you have already published your project files to the Web server, and now you want to update display you have changed.

The multiple display detection mechanism ensures that the Web page will be published properly for the source file. Disabling the **Multiple Display Support** may cause publishing of an incomplete web page, which may result in unexpected behavior. Therefore, it is suggested to disallow publishing of the Web page. You can do so by enabling the **Publish Display File Only** option.

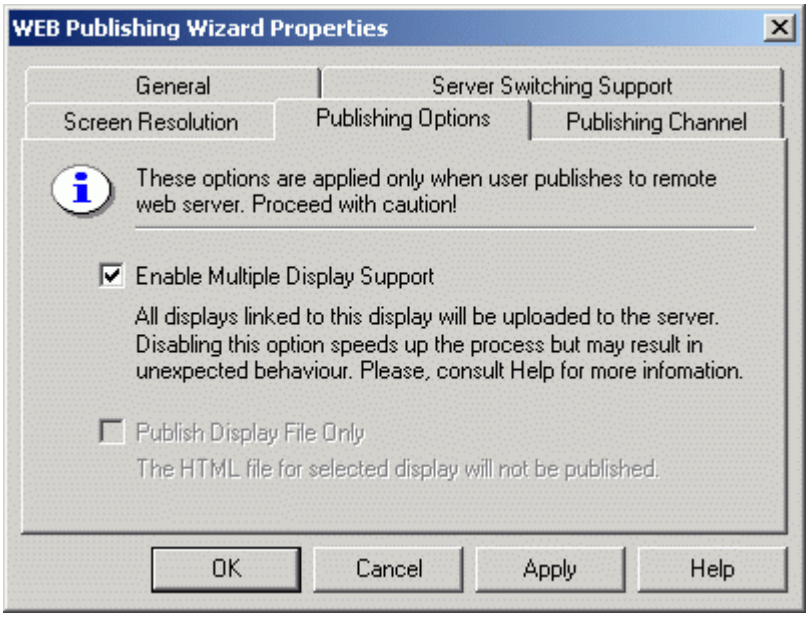

WEB Publishing Wizard Properties
General
Server Switching Support
Screen Resolution
Publishing Options
Publishing Channel
These options are applied only when user publishes to remote web server. Proceed with caution!
Enable Multiple Display Support
All displays linked to this display will be uploaded to the server. Disabling this option speeds up the process but may result in unexpected behaviour. Please, consult Help for more infomation.
Publish Display File Only
The HTML file for selected display will not be published.
OK
Cancel
Apply
Help

Figure 4.13. Publishing Options

Publishing Channel
Because publishing via Microsoft Posting Acceptor is no longer supported on Windows Server 2003, we were forced to find another way of transporting published files from a ProcessView development machine to a WebHMI server. It appeared that FTP would be the best choice.

Changing the Communication Channel
By default, the method that uses Posting Acceptor services is used for publishing files to a Web server. The **Publishing Channel** tab of the **Web Publishing Properties** dialog box, shown in **Figure 4.14,** enables you to change the way that files are uploaded to the Web server by selecting **FTP Channel.** This tells the Web Publishing Wizard to use FTP communications for uploading files instead of Posting Acceptor.

Note: Selecting **FTP Channel** does not affect the configuration of the main Web Publishing Wizard dialog; the settings are the same if you were publishing files using the default method (i.e. Posting Acceptor).

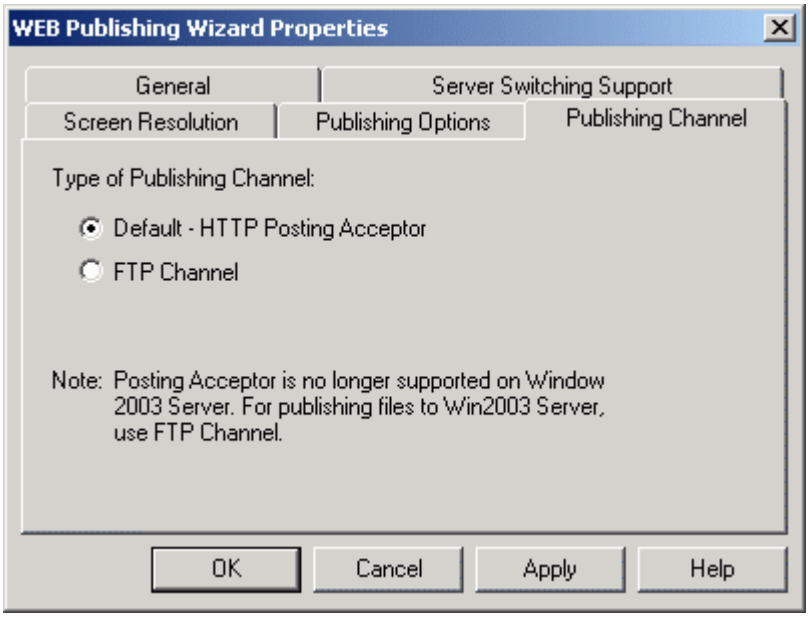


Figure 4.14. Publishing Channel

Configuring FTP Access
Just before uploading files to the Web server, you will be prompted to specify the FTP connection settings, as shown in **Figure 4.15.** Enter the correct FTP path, user name and password.

Note: For information about FTP site configuration, please see the application note called "Publishing to Windows 2003 Server," which is located on the ProcessView CD.

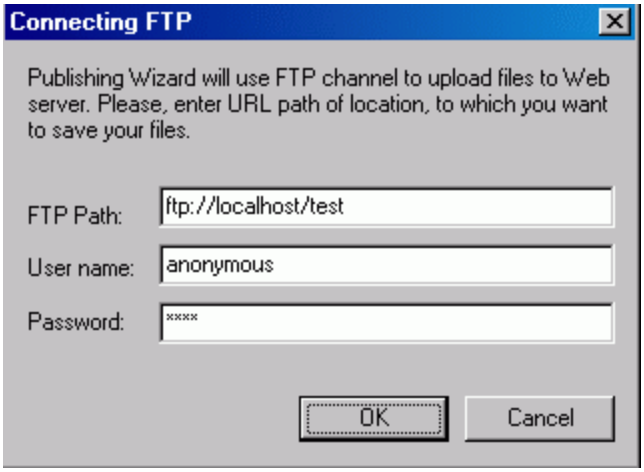


Figure 4.15. Uploading Files Via FTP

Viewing Locally Exported HTML Files

To view an HTML file that was exported to a local directory, simply browse to the local directory pathway and click on the file to open it in your Web browser. You can also open your Web browser and copy the directory pathway of the file into the **Address** field of the Web browser, as shown in **Figure 4.16.** A complete directory path is, for example, "C:\Documents and Settings\Administrator\My Documents\Web Publishing Example\ChemFood_BeanRoaster.htm." Be sure to include the name of the HTML file that you have exported and the file extension "htm."

Figure 4.16. Viewing the Exported HTML File in a Web Browser

Viewing Published HTML Files

Before connecting a client machine to a WebHMI Server, you must start the GenBroker Server on the WebHMI Server using ProcessView Tray. If the GenBroker Server is not started, then the client will not get proper licensing and security information. To start the GenBroker Server on the WebHMI Server:

1. Launch ProcessView Tray from the Windows **Start** menu by selecting **Programs > Smar ProcessView > Tools > ProcessView Tray.**

2. When ProcessView Tray opens, the triangle icon will appear in the Windows tool tray. Click on the triangle and select **GenBroker > Start** from the pop-up menu, as shown in **Figure 4.17.** This activates the GenBroker Server.

Figure 4.17. Starting the GenBroker Server on the Server Node

3. Once the GenBroker Server is running, you can view the HTML file that you have published to the Web server from any client PC's Web browser. To view the published HTML file, open your Web browser and copy the URL address of the file into the **Address** field of the Web browser, as shown in **Figure 4.18.** A complete address is, for example, "http://www.myserver.com/webhmi/Samples/ChemFood_BeanRoaster.htm." Be sure to include the name of the HTML file that you have published and the "htm" file extension.

The GraphWorX display file is referenced in the HTML code so the display can be viewed as a Web page. The client's Web browser simply downloads the HTML file in which the .gdf file is referenced. All the client needs is a Web browser; it is not necessary to have ProcessView installed on client. The Web page is downloaded from the Web server across the Internet/intranet and appears in the client's Web browser window. The display is in real-time, just as if you were viewing the runtime display in GraphWorX on the server machine; any live values (e.g. OPC tags) change dynamically in the display.

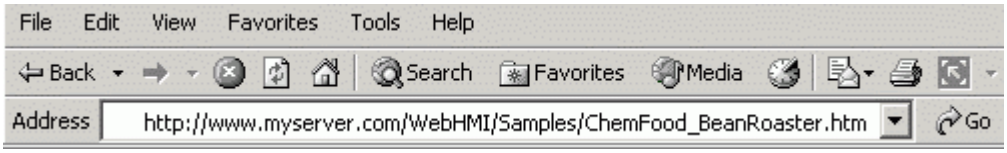


Figure 4.18. Viewing the Published HTML File in a Client Web Browser

Web Publishing in ProjectWorX

The Web Publishing Wizard can be accessed through the ProjectWorX console, allowing you to easily export/publish one or more GraphWorX (.gdf), AlarmWorX container (.a32), and TrendWorX container (.t32) display files within a project to a Web server. The following trees of each project include an option to publish a single display or multiple displays in the project:

- Applications/GraphWorX/Displays (as shown in the example figure below)
- Applications/AlarmWorX/Alarm Container
- Applications/TrendWorX/Trend Container

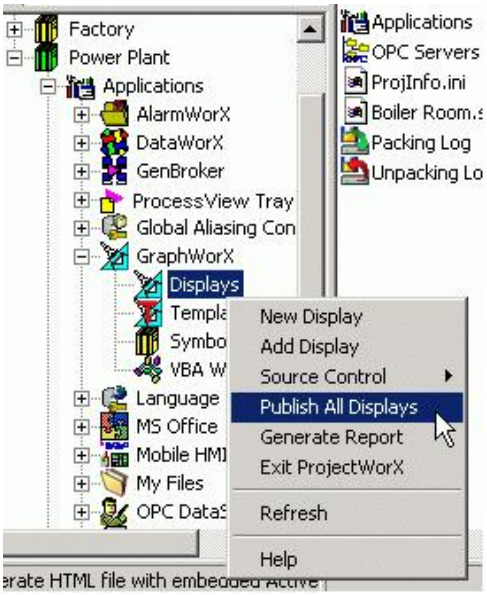


Figure 4.18. Publishing Displays From the ProjectWorX Console

The **WebHMI** tree under the **Applications** tree for each project in the ProjectWorX console, shown in the figure below, manages Web publishing for the entire project. From here you can right-click and publish HTML files to a Web server. The WebHMI tree includes the following categories:

- GraphWorX-Based HTML
- AlarmWorX-Based HTML
- TrendWorX-Based HTML
- Local HTML Files
- Published Documents

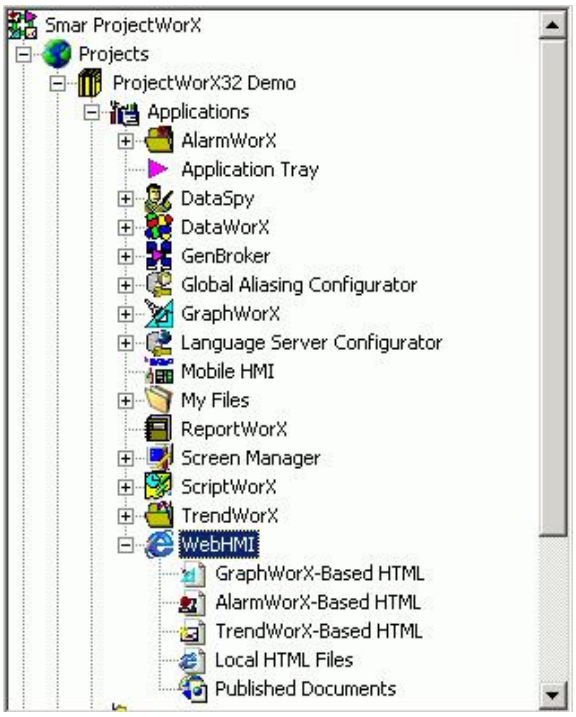


Figure 4.19. WebHMI Tree in ProjectWorX Console

Web Publishing From Alarm and Trend Viewer Files

ProjectWorX includes support for publishing TrendWorX container files (.t32) and AlarmWorX container files (.a32). The Web Publishing Wizard that is executed is identical to the wizard used for GraphWorX displays and identical to the wizard used when publishing directly form TrendWorX or AlarmWorX.

Unlike publishing GraphWorX displays, publishing trend and alarm files has the side effect of creating intermediate (.v32 and .awv) files. It is actually these files, and not the container files, that are referenced by the created HTML page. These files are contained in the **AlarmWorX Viewer Configuration** and **TrendWorX Viewer Configuration** trees, as shown in **Figure 4.20.** After the Web Publishing Wizard has been executed, these generated intermediate files are automatically added to these two trees. This is critical for ProjectWorX's ability to publish local HTML files with the intention of later publishing the HTML files from the local folders to a Web Site.

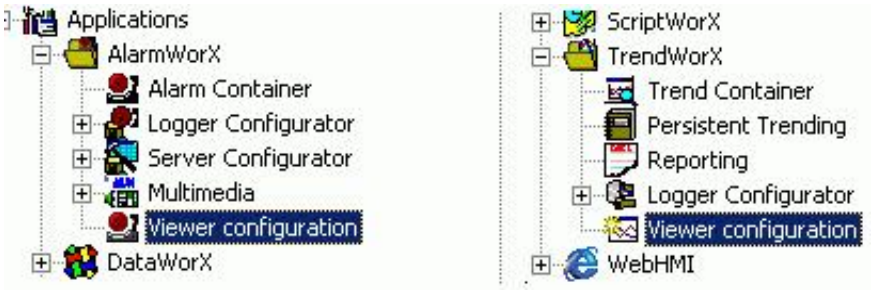


Figure 4.20. Viewer Configuration File Sections of the AlarmWorX and TrendWorX Trees

GraphWorX-Based HTML

The **GraphWorX-Based HTML** subtree of the WebHMI tree, shown in **Figure 4.21,** contains HTML files generated from exported GraphWorX displays. These HTML files have embedded code for accommodating a GraphWorX Viewer ActiveX Control. Any requested local copy of a GraphWorX-based HTML file generated from the Web Publishing Wizard is stored in this folder. (See the "Web Publishing Wizard" section below for more information on exporting and publishing HTML files from GraphWorX displays.) The ProjectWorX database maintains a record of which local GraphWorX display (.gdf) files each of these HTML files depends on. Documents in this folder can be added, removed, copied, pasted, opened, edited, added to Microsoft Visual SourceSafe, searched, replaced, and packed the same way items from other categories can.

Figure 4.21. GraphWorX-Based HTML Subtree

AlarmWorX-Based HTML

The **AlarmWorX-Based HTML** subtree of the WebHMI tree, shown in **Figure 4.22,** contains HTML files generated from exported AlarmWorX container displays. These HTML files have embedded code for accommodating an AlarmWorX Viewer ActiveX Control. Any requested local copy of an AlarmWorX-based HTML file generated from the Web Publishing Wizard is stored in this folder. (See the "Web Publishing Wizard" section below for more information on exporting and publishing HTML files from AlarmWorX displays.) The ProjectWorX database maintains a record of which local AlarmWorX container (.a32) files each of these HTML files depends on. Documents in this folder can be added, removed, copied, pasted, opened, edited, added to Microsoft Visual SourceSafe, searched, replaced, and packed the same way items from other categories can.

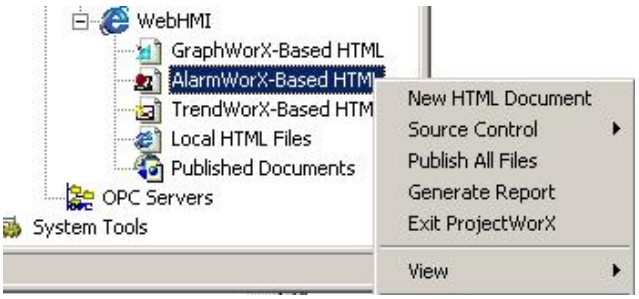


Figure 4.22. AlarmWorX-Based HTML Subtree

TrendWorX-Based HTML

The **TrendWorX-Based HTML** subtree of the WebHMI tree, shown in **Figure 4.23,** contains HTML files generated from exported TrendWorX container displays. These HTML files have embedded code for accommodating a TrendWorX Viewer ActiveX Control. Any requested local copy of an TrendWorX-based HTML file generated from the Web Publishing Wizard is stored in this folder. (See the "Web Publishing Wizard" section below for more information on exporting and publishing HTML files from TrendWorX displays.) The ProjectWorX database maintains a record of which local TrendWorX container (.t32) files each of these HTML files depends on. Documents in this folder can be added, removed, copied, pasted, opened, edited, added to Microsoft Visual SourceSafe, searched, replaced, and packed the same way items from other categories can.

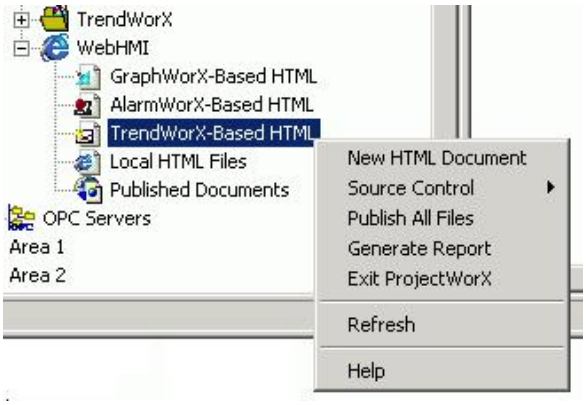


Figure 4.23. TrendWorX-Based HTML Subtree

Local HTML Files

The **Local HTML Files** subtree of the WebHMI tree stores basic HTML files that were not generated from the Web Publishing Wizard. Here you can add, delete, edit, and maintain various HTML files, as shown in **Figure 4.24.**

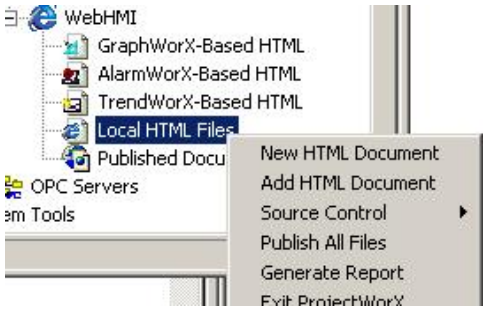


Figure 4.24. Local HTML Files Subtree

Published Documents

The **Published Documents** subtree of the WebHMI tree, shown in **Figure 4.25,** maintains a list of all HTML files that have been exported/published from ProjectWorX.

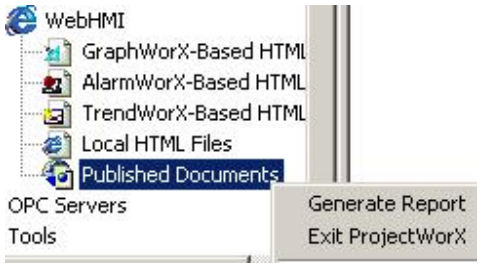


Figure 4.25. Published Documents Subtree

In the right-hand pane of the ProjectWorX Console, each item is listed as a URL with the date it was published, as shown in **Figure 4.26.** For information about publishing files from ProjectWorX, please see "Using the Web Publishing Wizard."

URL	Publish Date
http://www.webhmi.com/backup/webHMI/testActiveX.html	7/5/2002 6:34:...
http://www.webhmi.com/backup/webHMI/NewFile.htm	7/5/2002 6:50:...
http://www.webhmi.com/backup2/webHMI/NewFile.htm	7/5/2002 6:50:...
http://world/Publish_test/LinkTest2.gdf.htm	7/9/2002 10:40...

Figure 4.26. List of All Documents Published From ProjectWorX

Launching the Web Publishing Wizard in ProjectWorX

The Web Publishing Wizard in ProjectWorX, shown in **Figure 4.27,** performs two basic operations:

1. The Wizard creates HTML files based on user-specified GraphWorX display (.gdf) files, AlarmWorX container (.a32) files, and TrendWorX container (.t32) files.
2. The Wizard then either "exports" (saves) the HTML files to a user-specified directory on the local drive and/ or "publishes" (uploads) the HTML file to a user-specified Web server URL address (i.e. over the Internet or an intranet).

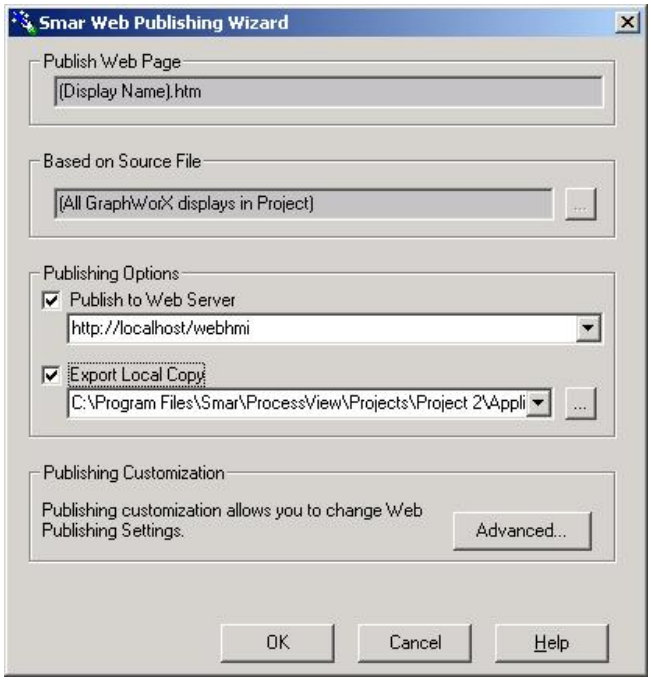


Figure 4.27. Web Publishing Wizard in ProjectWorX Console

You can open the Web Publishing Wizard dialog from the ProjectWorX console in the following ways:

- Select the project root and then choose **Publish Wizard** from the **Tools** menu, as shown in **Figure 4.28.**

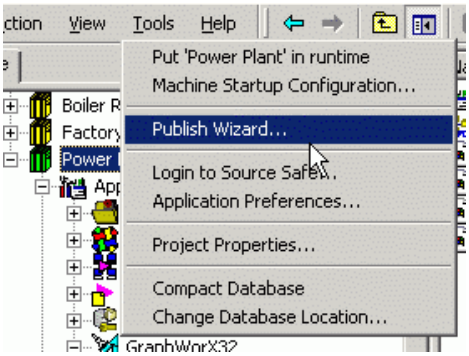


Figure 4.28. Opening the Publishing Wizard From the Tools Menu

Choose the type of file to publish, as shown in **Figure 4.29.** In this case, both the **Publish Web Page** field and the **Based on Source File** field in the Web Publishing Wizard dialog are read-only and contain the text "(Display Name).htm" and "(All [GraphWorX, AlarmWorX, or TrendWorX] Displays in Project)," respectively. The browse (**…**) button is also disabled.

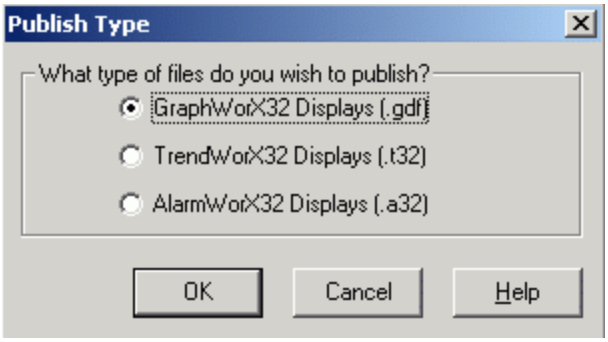


Figure 4.29. Choosing a File Type to Publish

- Right-click on the **GraphWorX/Displays** (or the **AlarmWorX/Alarm Container** tree or the **TrendWorX/Trend Container**) tree and select **Publish All Displays,** as shown in **Figure 4.30.** In this case, both the **Publish Web Page** field and the **Based on Source File** field in the Web Publishing Wizard dialog are read-only and contain the text "(Display Name).htm" and "(All [GraphWorX, AlarmWorX, or TrendWorX] Displays in Project)," respectively. The browse (**…**) button is also disabled.

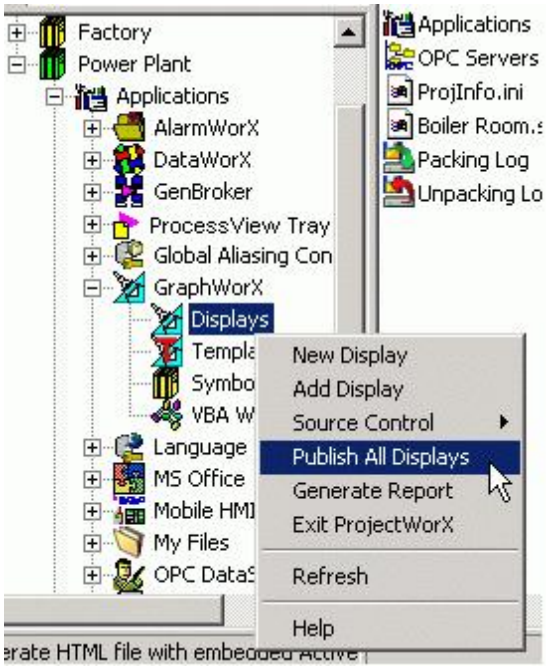


Figure 4.30. Publishing All Displays in a Project

- Right-click on a GraphWorX (.gdf), AlarmWorX (.a32), or TrendWorX (.t32) file and select **Publish to HTML,** as shown in **Figure 4.31.** In this case, the **Based on Source File** field in the Web Publishing Wizard dialog is read-only and contains the name of the selected display file. The browse (**…**) button is also disabled.

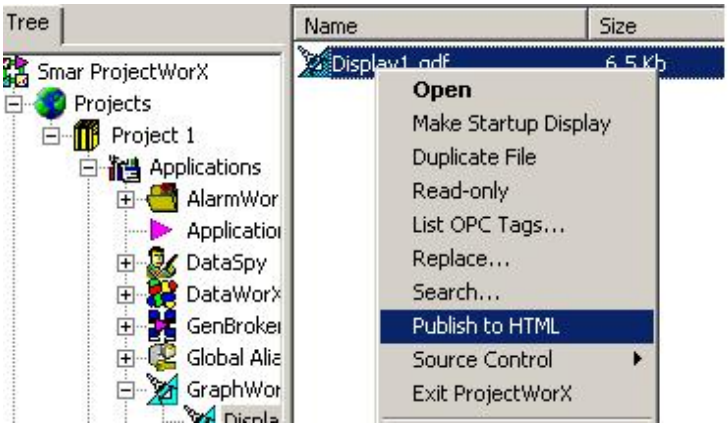


Figure 4.31. Publishing a Display File

- Right-click on the **Applications/WebHMI/GraphWorX-Based HTML** (or **AlarmWorX-Based HTML** or **TrendWorX-Based HTML)** tree and select **New HTML Document,** as shown in **Figure 4.32.** In this case, you need to specify a GraphWorX display file in the **Based on Source File** field by browsing for the file or by typing in the file name. You also need to specify the name of the new HTML file in the **Publish Web Page** field.

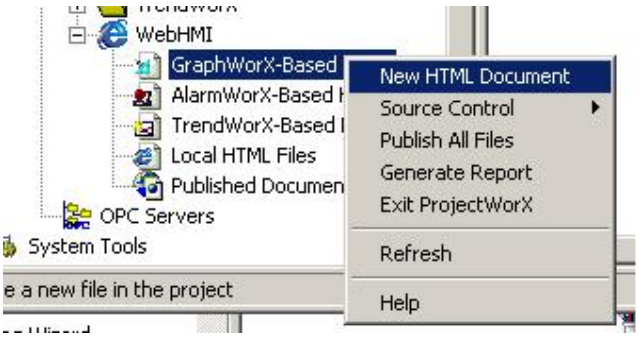


Figure 4.32. Creating a New HTML File in a Project

Export and Publish Options in ProjectWorX

The Web Publishing Wizard in the ProjectWorX console, shown in **Figure 4.33,** generates an HTML file based on one or more user-specified GraphWorX display (.gdf) files, AlarmWorX container (.a32) files, and TrendWorX container (.t32) files. Then the Wizard either "exports" (saves) the HTML file to a user-specified directory on the local drive or "publishes" (uploads) the HTML file to a user-specified Web server URL address (i.e. over the Internet or an intranet).

You have the following options when using the Web Publishing Wizard for a project:

- Export and/or publish all files in the project.
- Export and/or publish a single file in the project.
- Export a file locally
- Publish a file to a Web server.

Note: Both the **Publish to Web Server** and **Export Local Copy** actions can be performed at the same time.

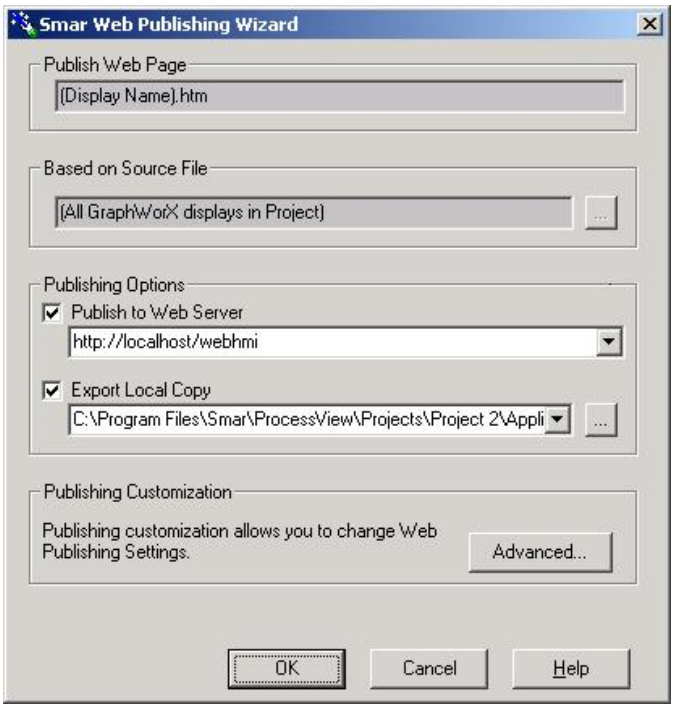


Figure 4.33. Web Publishing Wizard in ProjectWorX Console

Exporting and Publishing Multiple Display Files in ProjectWorX

In ProjectWorX, you can export and/or publish multiple display files using the Web Publishing Wizard.

To export/publish all displays in a project:

1. Select the project root and then choose **Publish Wizard** from the **Tools** menu, or right-click on the **Applications/GraphWorX/Displays** tree (or the **AlarmWorX/Alarm Container** tree or the **TrendWorX/Trend Container**) and select **Publish All Displays,** as shown in **Figure 4.34.**

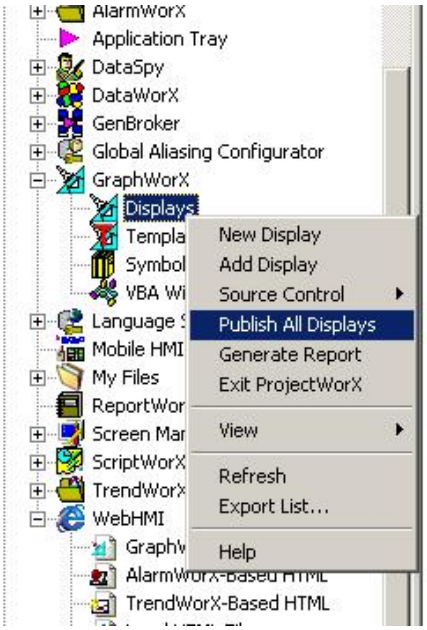


Figure 4.34. Launching the Web Publishing Wizard

2. This launches the Web Publishing Wizard, as shown in **Figure 4.35.** In this case, both the **Publish Web Page** field and the **Based on Source File** field in the Web Publishing Wizard dialog are read-only and contain the text "(Display Name).htm" and "(All [GraphWorX, AlarmWorX, or TrendWorX] Displays in Project)," respectively. The browse (**…**) button is also disabled.

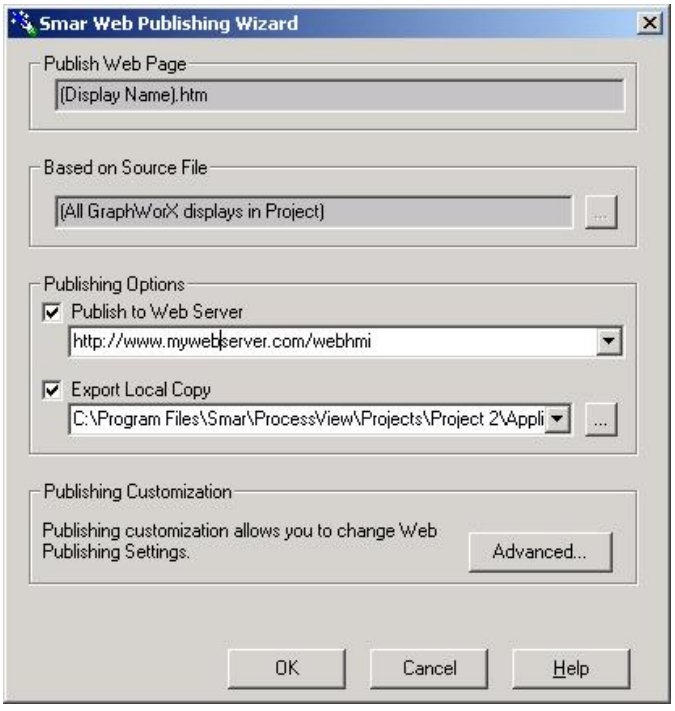


Figure 4.35. Publishing All Displays in a Project

Exporting and Publishing Individual Display Files in ProjectWorX

In ProjectWorX, you can export and/or publish individual files using the Web Publishing Wizard.

To export/publish a single display in a project:

1. Right-click on a GraphWorX (.gdf) file, an AlarmWorX container (.a32) file, or a TrendWorX container (.t32) file and select **Publish to HTML,** as shown in **Figure 4.36.**

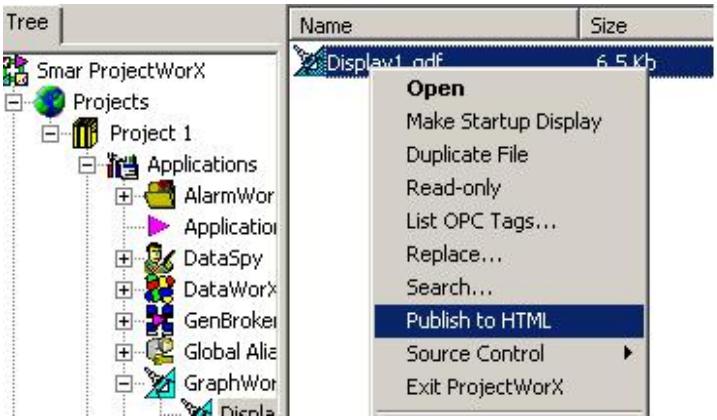


Figure 4.36. Launching the Web Publishing Wizard

2. This launches the Web Publishing Wizard, as shown in **Figure 4.37.** In this case, the **Based on Source File** field in the Web Publishing Wizard dialog is read-only and contains the name of the selected display file. The browse (**...**) button is also disabled.

3. In the **Publish Web Page** field of the Web Publishing Wizard, specify the name of the HTML file that will be created. The file name is filled in by default, but you can give the HTML file a different name.

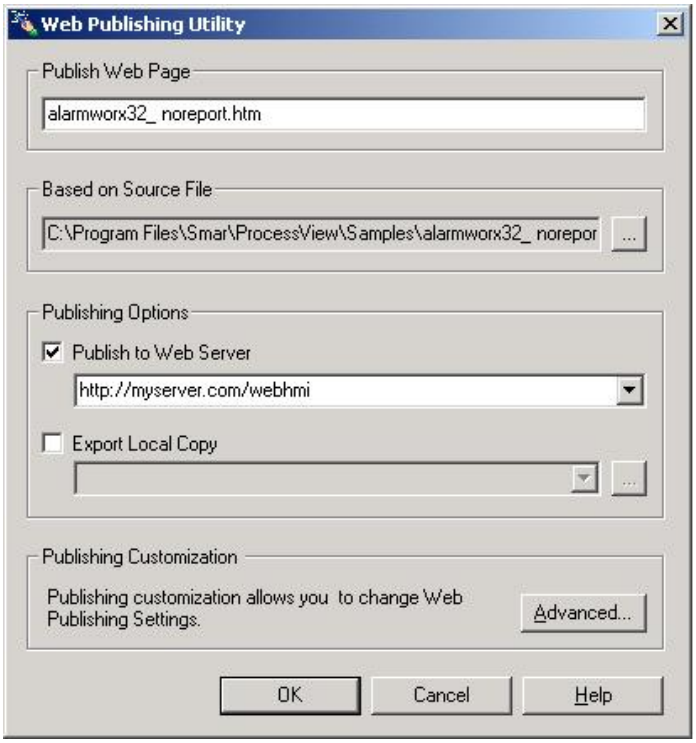


Figure 4.37. Publishing a Display in a Project

Note: You can also right-click on the **Applications/WebHMI/GraphWorX-Based HTML** tree (or the **AlarmWorX-Based HTML** tree or the **TrendWorX-Based HTML** tree) and select **New HTML Document,** as shown in **Figure 4.38,** to export/publish a display. This launches the Web Publishing

Wizard. In this case, you need to specify a display file in the **Based on Source File** field by browsing for the file or by typing in the file name. You also need to specify the name of the new HTML file in the **Publish Web Page** field.

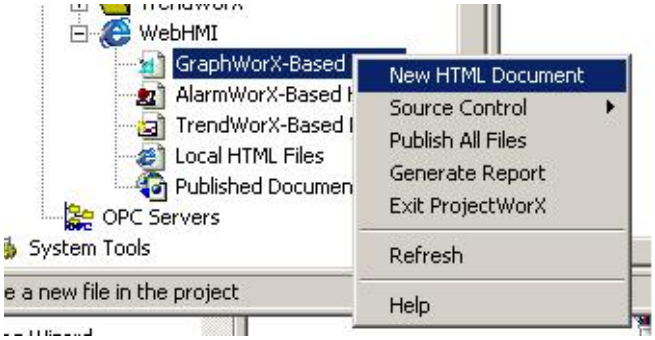


Figure 4.38. Creating a New HTML File in a Project

Exporting a Display File Locally in ProjectWorX

To export a display file to a directory on the local drive:

1. Right-click on the **Applications/WebHMI/GraphWorX-Based HTML** tree (or the **AlarmWorX-Based HTML** tree or the **TrendWorX-Based HTML** tree) and select **New HTML Document,** as shown in **Figure 4.39.**

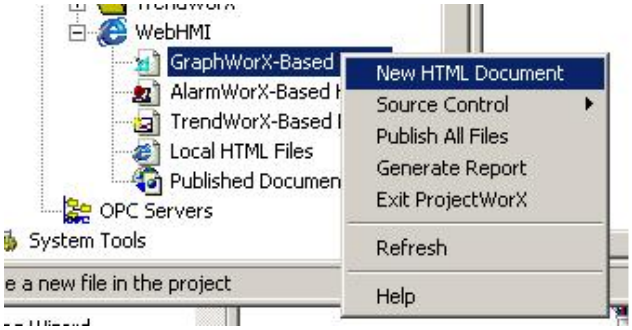


Figure 4.39. Launching the Web Publishing Wizard

2. This launches the Web Publishing Wizard, as shown in **Figure 4.40.** In the **Based on Source File** field of the Web Publishing Wizard, you must specify the name of the display file to be exported, as shown in the figure below. To choose a display, click the **…** button to the right to browse for a file. Select a file and then click **Open.** The directory path and the file name are shown in the text field.

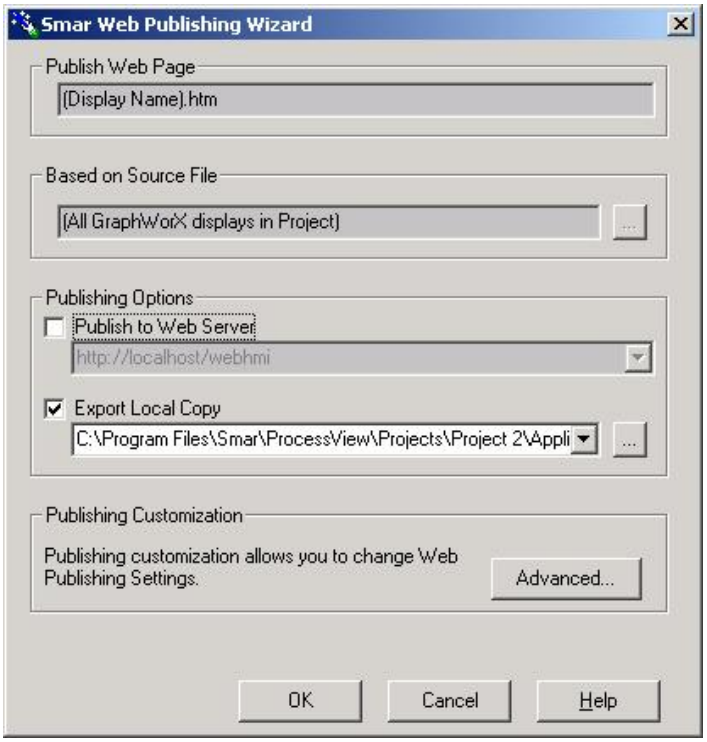


Figure 4.40. Exporting a Display File Locally in ProjectWorX

3. In the **Publish Web Page** field of the Web Publishing Wizard, specify the name of the HTML file that will be created.

4. In the **Publishing Options** field, the **Export Local Copy** check box is checked by default. Specify the local directory path name to which you want to export the HTML file. You can select a recently used path from the drop-down list, or click the **…** button to browse for a destination directory. Select the directory and click **OK.** The local directory pathway you selected appears in the **Export Local Copy** field of the Web Publishing Wizard.

Note: By default, ProjectWorX saves the HTML file to the (local) **Applications/WebHMI/GraphWorX-Based HTML** tree (or the **AlarmWorX-Based HTML** tree or the **TrendWorX-Based HTML** tree). From there, the HTML file can be edited and published again in the future.

5. Click the **OK** button to generate the HTML file. The new HTML file is saved to the local directory pathway specified in the **Export Local Copy** field.

Note: Both the **Publish to Web Server** and **Export Local Copy** actions can be performed at the same time.

Publishing a Display File to a Web Server in ProjectWorX

To publish the HTML file to a directory on a Web server (i.e. over the Internet or an intranet):

1. Right-click on the **Applications/WebHMI/GraphWorX-Based HTML** tree (or the **AlarmWorX-Based HTML** tree or the **TrendWorX-Based HTML** tree) and select **New HTML Document,** as shown in **Figure 4.41.**

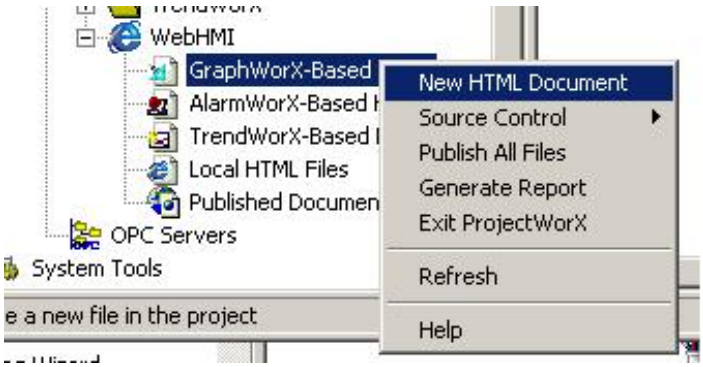


Figure 4.41. Launching the Web Publishing Wizard

2. This launches the Web Publishing Wizard, as shown in **Figure 4.42.** In the **Based on Source File** field of the Web Publishing Wizard, you must specify the name of the file to be published, as shown in Figure 4.42. To choose a display, click the **…** button to the right to browse for a file. Select a file and then click **Open.** The directory path and the file name are shown in the text field.

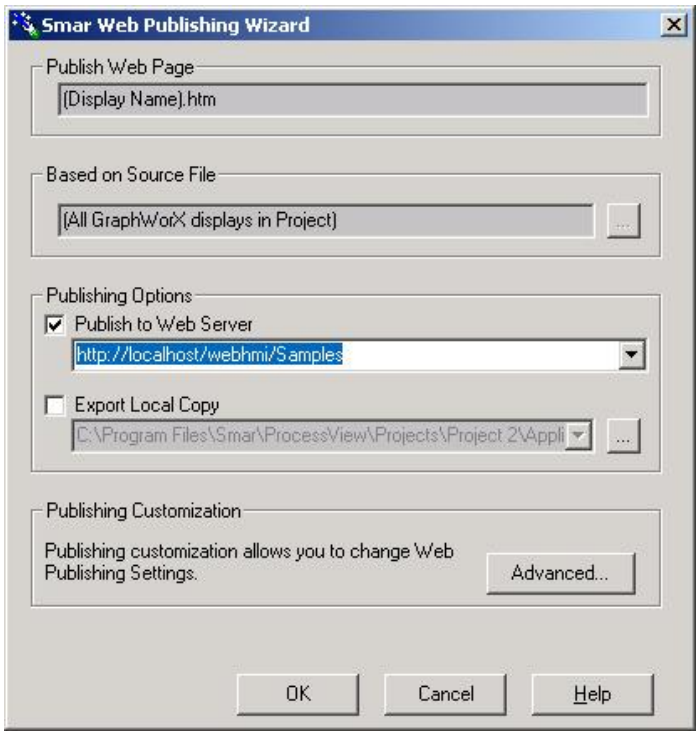


Figure 4.42. Publishing a Display File to a Web Server

3. In the **Publish Web Page** field of the Web Publishing Wizard, specify the name of the HTML file that will be created.

4. In the **Publishing Options** field, check the **Publish to Web Server** check box and type in the URL address of your Web server with the complete directory indicating where you want to publish the HTML file, as shown in Figure 4.42. In the sample Web server URL address shown above ("http://www.myserver.com/WebHMI/Samples"), the various components are:

- **http://www.myserver.com:** IP address (server name) of the WebHMI Server
- **WebHMI:** Name of the WebHMI server root directory
- **Samples:** Name of the directory on the server to which the HTML file will be saved

5. Click the **OK** button to generate the HTML file. The new HTML file is uploaded to the Web server and then saved to the URL address path specified in the **Publish to Web Server** field.

Note: By default, ProjectWorX also saves the HTML file to the (local) **Applications/WebHMI/GraphWorX-Based HTML** tree (or the **AlarmWorX-Based HTML** tree or the **TrendWorX-Based HTML** tree). From there, the HTML file can be edited and published again in the future.

Note: Both the **Publish to Web Server** and **Export Local Copy** actions can be performed at the same time.

Publishing Customization Options

The Web Publishing Wizard contains some customization options for exporting/publishing HTML files. Clicking the **Advanced** button on the Web Publishing Wizard dialog opens the **Web Publishing Properties** dialog box, which contains the following tabs:

- General Settings
- Server Switching Support
- Screen Resolution Settings
- Publishing Options
- Publishing Channel

General Settings

The **General** tab of the **Web Publishing Properties** dialog box, shown in **Figure 4.43,** allows you to specify the default URL address for the WebHMI Server. The **General** tab is visible only when **Export Local Copy** is the sole option selected on the Web Publishing Wizard dialog box. If **Publish to Web Server** is selected on the Web Publishing Wizard dialog box, then the **General** tab is hidden and you won't need to configure it.

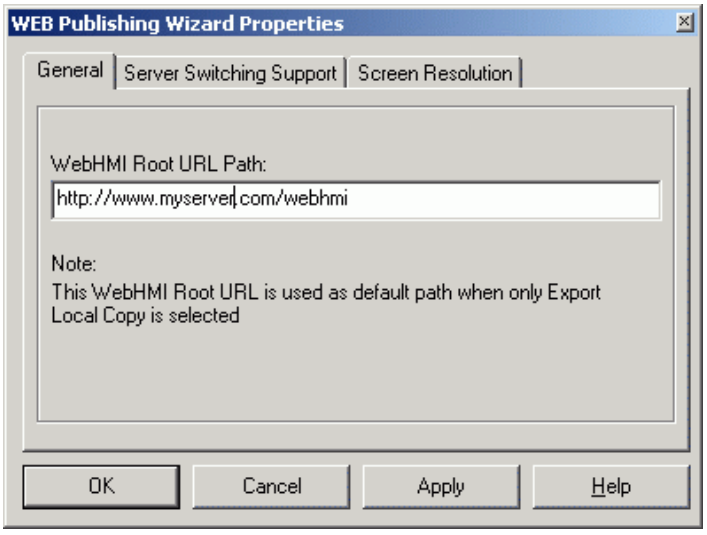


Figure 4.43. General Settings

Server Switching Support

The **Server Switching Support** tab of the **Web Publishing Properties** dialog box, shown in **Figure 4.44,** allows you to enable or disable GenBroker support for the exported/published HTML file. Here you can specify which GenBroker configuration file (.gbx) will be activated. The GenBroker configuration file establishes the settings for OPC data communications between the clients and the Web server.

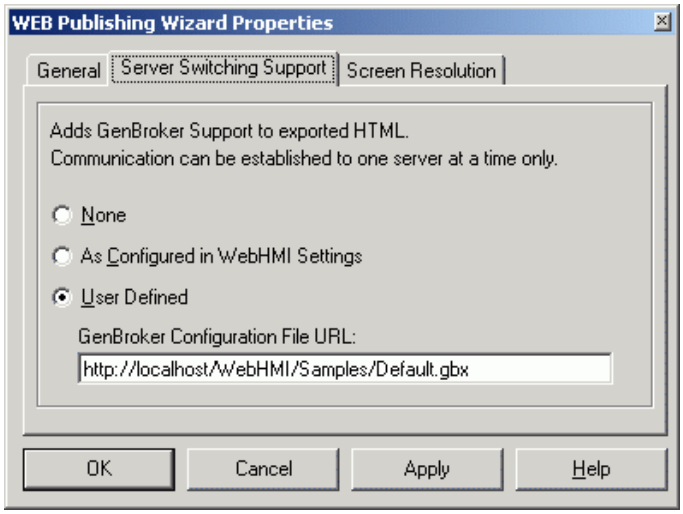


Figure 4.44. Server Switching Support Settings

There are three available options for GenBroker support:

- **None:** GenBroker support is disabled. If this option is selected, all the communication will default to the local node. This option should be checked when the Web pages are meant to be used only from the local node. If this option is checked, Web pages may not work on a remote node, unless the remote node has ProcessView installed and GenBroker is properly configured to communicate with the WebHMI server.
- **As Configured in WebHMI Settings:** This setting uses the default URL address of the GenBroker configuration (.gbx) file as specified in the WebHMI installation. This is the recommended option for non-expert users.
- **User Defined:** When this option is selected, the **GenBroker Configuration File URL** field becomes available, allowing you to specify the URL address of the GenBroker configuration (.gbx) file. This option gives the maximum flexibility and should be used by expert users only.

Note: For information about configuring GenBroker settings, please see the GenBroker Configurator Help documentation.

Screen Resolution Settings

The **Screen Resolution** tab of the **Web Publishing Properties** dialog box, shown in **Figure 4.45,** determines the screen resolution and size of the GraphWorX Viewer ActiveX control, which is referenced in the generated HTML file and then downloaded to a client PC when the HTML file is viewed in the client's Web browser. You can specify the screen size in the **Width** and **Height** fields in terms of pixels (px) or percentage (%). Click the **Resolutions** button to select from a list of standard screen resolutions, as shown in Figure 4.45.

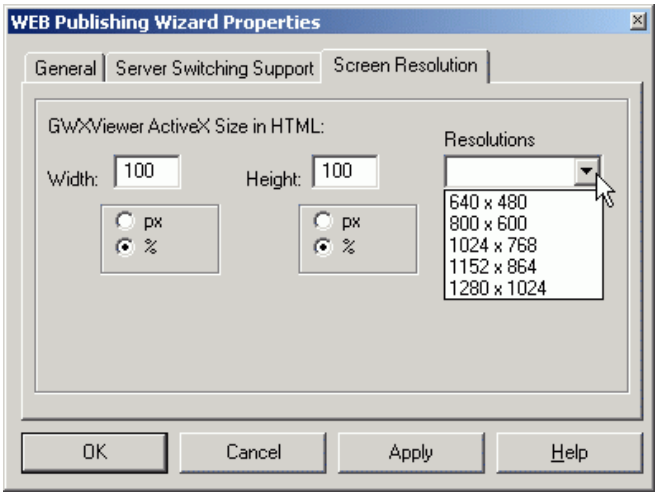


Figure 4.45. Screen Resolution Settings

Publishing Options

The **Publishing Options** tab of the **Web Publishing Properties** dialog box, shown in **Figure 4.46,** enhances the publishing process. By default, the Web Publishing Wizard provides support for related files detection (for more details, see the **Multiple Display Support** section). The multiple display detection could be a rather lengthy process. You can speed up the process by disabling the **Enable Multiple Display Support** option. This is especially useful, if you have already published your project files to the Web server, and now you want to update display you have changed.

The multiple display detection mechanism ensures that the Web page will be published properly for the source file. Disabling the **Multiple Display Support** may cause publishing of an incomplete web page, which may result in unexpected behavior. Therefore, it is suggested to disallow publishing of the Web page. You can do so by enabling the **Publish Display File Only** option.

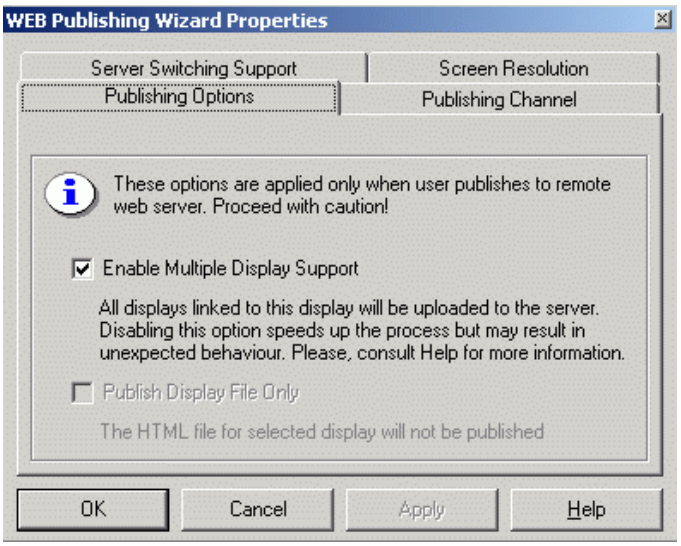

WEB Publishing Wizard Properties
Server Switching Support
Screen Resolution
Publishing Options
Publishing Channel
These options are applied only when user publishes to remote web server. Proceed with caution!
Enable Multiple Display Support
All displays linked to this display will be uploaded to the server. Disabling this option speeds up the process but may result in unexpected behaviour. Please, consult Help for more information.
Publish Display File Only
The HTML file for selected display will not be published
OK
Cancel
Apply
Help

Figure 4.46. Publishing Options

Publishing Channel

Because publishing via Microsoft Posting Acceptor is no longer supported on Windows Server 2003, we were forced to find another way of transporting published files from a ProcessView development machine to a WebHMI server. It appeared that FTP would be the best choice.

Changing the Communication Channel

By default, the method that uses Posting Acceptor services is used for publishing files to a Web server. The **Publishing Channel** tab of the **Web Publishing Properties** dialog box, shown in **Figure 4.47,** enables you to change the way that files are uploaded to the Web server by selecting **FTP Channel.** This tells the Web Publishing Wizard to use FTP communications for uploading files instead of Posting Acceptor.

Note: Selecting **FTP Channel** does not affect the configuration of the main Web Publishing Wizard dialog; the settings are the same if you were publishing files using the default method (i.e. Posting Acceptor).

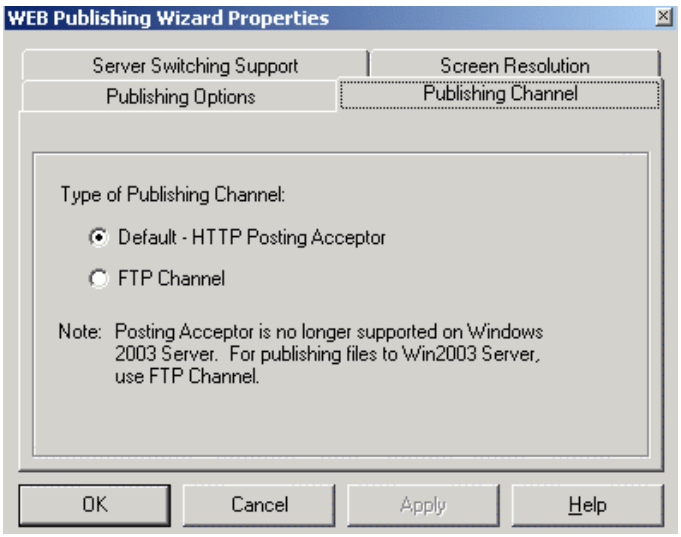


Figure 4.47. Publishing Channel

Configuring FTP Access

Just before uploading files to the Web server, you will be prompted to specify the FTP connection settings, as shown in **Figure 4.48.** Enter the correct FTP path, user name and password.

Note: For information about FTP site configuration, please see the application note called "Publishing to Windows 2003 Server," which is located on the ProcessView CD.

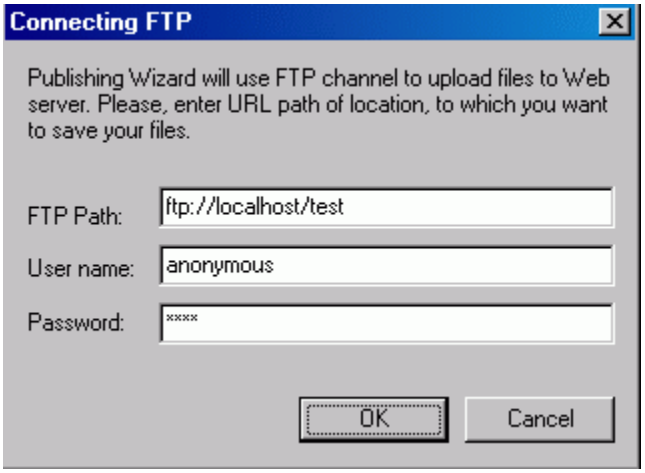


Figure 4.48. Uploading Files Via FTP

Publishing Files at a Later Time

The Web Publishing Wizard, ProjectWorX saves the generated HTML file to the (local) **Applications/WebHMI/GraphWorX-Based HTML** tree (or the **AlarmWorX-Based HTML** tree or the **TrendWorX-Based HTML** tree). The advantage of storing a local copy of an exported HTML file is that it can be edited, packed, moved, unpacked, and then published again at any time. This saves you the burden of having to re-export the same file to a new HTML file. You can use this option in any of the following ways:

- Right-click on any HTML file in the **Applications/WebHMI/GraphWorX-Based HTML** tree (or the **AlarmWorX-Based HTML** tree or the **TrendWorX-Based HTML** tree) and select **Publish to Web Server** from the pop-up menu, as shown in **Figure 4.49.**
- Right-click on any HTML file in the **Applications/WebHMI/Local HTML Files** tree and select **Publish to Web Server** from the pop-up menu.

Note: Selecting **Edit HTML** from the pop-up menu opens the HTML file in the default HTML editor.

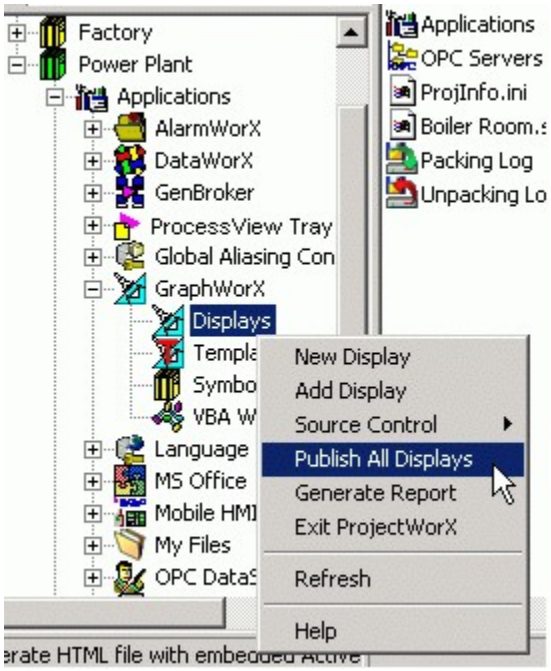


Figure 4.49. Publishing a Locally Stored HTML File

- Right-click on the **GraphWorX-Based HTML Files** tree (or the **AlarmWorX-Based HTML** tree or the **TrendWorX-Based HTML** tree) and select **Publish All Files** from the pop-up menu, as shown in **Figure 4.50.**
- Right-click on the **Local HTML Files** tree and select **Publish All Files** from the pop-up menu.
- Right-click on the **WebHMI** tree and select **Publish All Files** from the pop-up menu. This publishes all HTML files in the WebHMI tree.

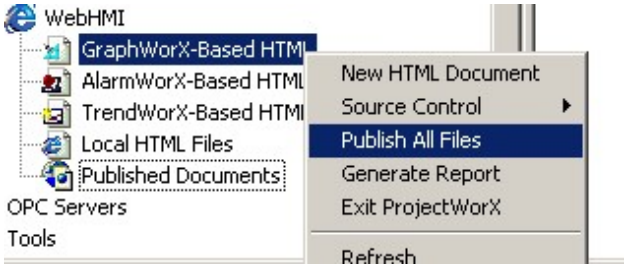


Figure 4.50. Publishing All Locally Stored HTML Files

All of these actions open the **Publish HTML File to Web Site** dialog box, as shown in **Figure 4.51.** The **Qualified WWW Folder** field in this dialog is identical to the **Publish to Web Server** field in the Web Publishing Wizard dialog and is stored in the same place in the registry if the action is successful. This specifies the location of the Web server to which the HTML file will be published. If the export/publish action is successful, a "publish" action on an HTML file in the **Local HTML Files** tree simply uploads the HTML file to the Web server. A "publish" action on an HTML file in the **GraphWorX-Based HTML Files** tree (or the **AlarmWorX-Based HTML** tree or the **TrendWorX-Based HTML** tree) uploads both the HTML file and its attached display file to the Web server.

Figure 4.51. Specifying a Web Server Location for HTML Publishing

Web Publishing Log

When exporting/publishing only one file from a display to HTML, you are alerted if anything goes wrong in the process. If, however, multiple files are being exported/published simultaneously, a log is generated so the operation will not be halted with each file that cannot be exported/published. After a batch publishing operation from the **GraphWorX-Based HTML Files** tree (or the **AlarmWorX-Based HTML** tree or the **TrendWorX-Based HTML** tree), a log file appears showing the success or failure of each attempt to publish the files. If this log file is needed at a later time, it can be found in the project root, as shown in **Figure 4.52.**

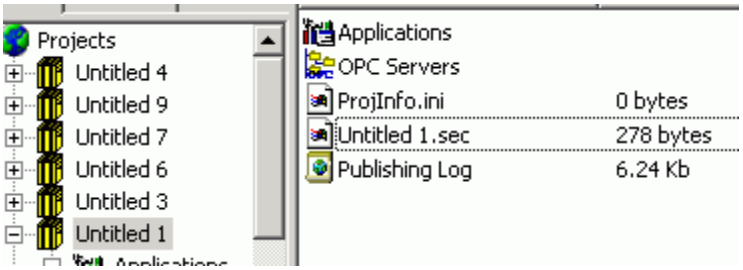


Figure 4.52. Location of Web Publishing Log File

Chapter 5

WEBHMI CLIENT CONFIGURATION

WebHMI Client/Server Architecture

GenBroker acts as a bridge that links WebHMI clients to the WebHMI server over the Internet. The WebHMI clients are used to view Web pages downloaded from the WebHMI server through a Web browser (Microsoft Internet Explorer). As shown in **Figure 5.1,** GenBroker enables communication between the Web clients and the WebHMI server over TCP/IP.

Note: For complete information about configuring GenBroker communications, please refer to the GenBroker Help documentation.

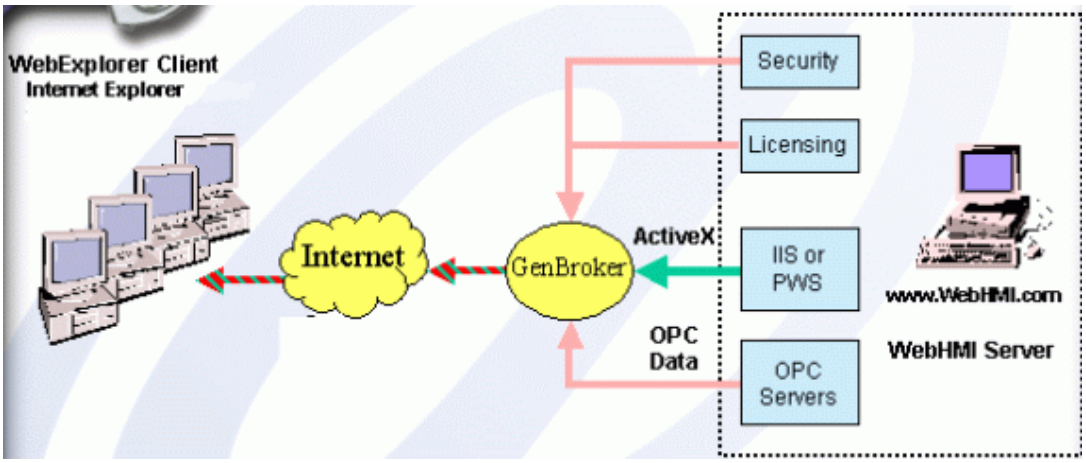


Figure 5.1. WebHMI Client/Server Architecture

Configuring GenBroker

The GenBroker Configurator is a tool that enables you to customize your network architecture. The GenBroker Configurator allows you to:

- Specify the communication method.
- Configure the GenBroker Client.
- Configure the GenBroker Server.
- Assign nodes to direct channels.
- Assign mediator nodes for indirect channels.
- Specify the default channel.
- Add and remove nodes or IP addresses.
- Configure channel properties.
- Specify the location of primary and redundant remote servers.

The configuration settings are saved in a GenBroker configuration (.gbx) file. You can store multiple configuration files. Configuration files tell GenBroker Server how to communicate when it interacts with various clients.

Note: The .gbx configuration file is the preferred format introduced in ProcessView version 7.x and used in subsequent versions. The .gbc format used in ProcessView 6.x and prior versions only contained primary nodes of security and licensing servers as well as definitions of channels, while the .gbx format contains the primary and backup nodes for all kinds of ICONICS servers. The .gbx file is stored in XML, while the .gbc is binary. Unlike a .gbc file,.a gbx file contains all options formerly stored in the system registry. This allows WebHMI clients to download the options defined on the WebHMI Server. In addition, .gbx contains definitions for the new features, such as shortcuts. The .gbc file is used for backward compatibility with version 6.x. It is recommended that you use the .gbx format in version 7.x and version 8.x.

Note: GenBroker has already been configured prior to installation. Configure (or reconfigure) GenBroker only if you wish, for example, to change servers.

GenBroker is configured for both the client side and the server side. For WebHMI, the GenBroker configuration (.gbx) file used by the client resides on the server side. Once the client connects to the server, the file is downloaded from the server. ProcessView clients have their own .gbx file. Note that a GenBroker Server does not need to be a GenBroker client or a Web server.

Note: For complete information about configuring GenBroker communications and GenBroker Server, please refer to the GenBroker Help documentation. For information about WebHMI client configuration, please see the "Configuring GenBroker for the Client Side" section below.

Configuring GenBroker for the Client Side

GenBroker is configured for both the client side and the server side. Here we are concerned with WebHMI client configuration. For information on how to configure GenBroker for the server side, and for example GenBroker configurations, please see the GenBroker Help documentation.

WebHMI uses a configuration (.gbx) file for GenBroker. The GenBroker configuration file used by the client resides on the server side. Once the client connects to the server, the .gbx file will be downloaded from the server.

The first step in the WebHMI client configuration is to specify which .gbx file to use. You can use a default file provided in the WebHMI installation, or you can you can create a new file. To work with a .gbx file, you must start the WebHMI client configuration from the Windows **Start** menu by selecting **Programs > ICONICS WebHMI > Client Configuration.**

This opens the **WebHMI Configurator** dialog box, as shown in Figure 5.2. To use the default configuration file, click the **...** button to the right of the Edit the **GenBroker Configuration** field. This allows you to select a .gbx file. Browse to "InetPub\wwwroot\WebHMI\Samples" and select **Default.gbx.**

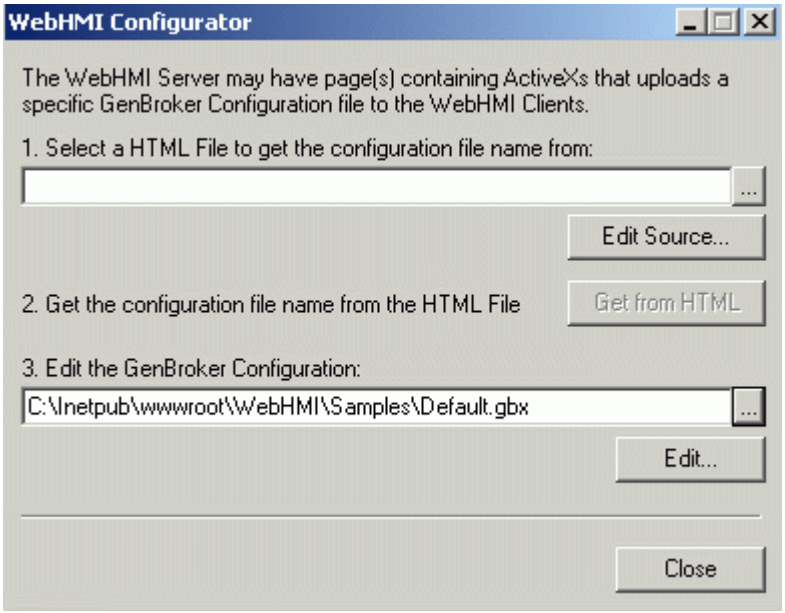

WebHMI Configurator
The WebHMI Server may have page(s) containing ActiveXs that uploads a specific GenBroker Configuration file to the WebHMI Clients.
1. Select a HTML File to get the configuration file name from:
...
Edit Source...
2. Get the configuration file name from the HTML File
Get from HTML
3. Edit the GenBroker Configuration:
C:\Inetpub\wwwroot\WebHMI\Samples\Default.gbx
...
Edit...
Close

Figure 5.2. WebHMI Client Configuration

The file pathway will appear in the text box, as shown in the figure above. You do not have to reconfigure the default file except to:

- Redirect channels of a WebHMI server.
- Add a channel for a new node whose resident OPC server provides data to client PC browsers.

If you want to set up a different or more advanced client/server communication architecture, you can either reconfigure the "default.gbx" file or create a new .gbx file.

Retrieving a GenBroker Configuration File From a Web Page
Alternatively, you can search for a GenBroker configuration file in an HTML file. In many cases, .gbx files are referenced in the source code of HTML files. Click the **...** button to the right of the **Web Page (HTML File)** field on the **WebHMI** tab of the **GenBroker Properties** dialog box. Browse to select an HTML file, and then click the **Open** button. The HTML file pathway will now be displayed in the **Web Page (HTML File)** text box. In addition, the **Edit Source** button will be enabled. Clicking **Edit Source** allows you to view the source code for the selected HTML file. Once you have selected the HTML file, click the **Get From HTML** button to retrieve the .gbx file that is referenced in the HTML source code.

Editing the Default Configuration File
To reconfigure the default configuration, click the **Edit** button on the **WebHMI Configurator** dialog box. This opens the GenBroker Configurator, as shown in **Figure 5.3.** In the example, the IP address "99.99.99.99" is the default mediator node and is the node assigned to the Security Server and the License Server. You can add, remove, or reconfigure nodes using the commands and functions described in the sections above.

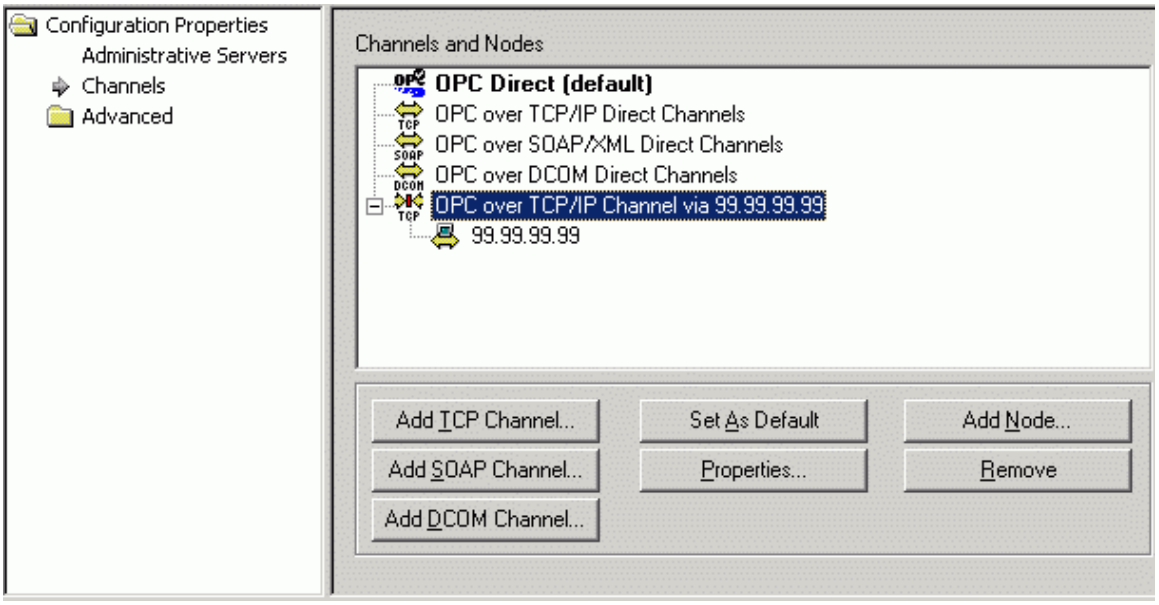


Figure 5.3. Editing the Default GenBroker Configuration

Creating a New GenBroker Configuration File
To create a new GenBroker configuration file, select **New** from the **File** menu in the GenBroker Configurator, as shown in **Figure 5.4.** Save the new .gbx file by selecting **Save** from the **File** menu. You can add and configure nodes using the commands and functions described in the GenBroker Help documentation. To view the GenBroker Help, select **Help Topics** from the **Help** menu.

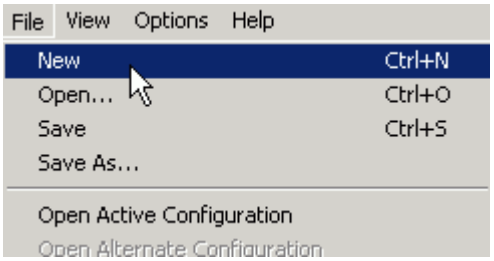


Figure 5.4. Creating a New GenBroker Configuration File

Example GenBroker Configuration

This section shows an example GenBroker configuration for the sample network setup shown in **Figure 5.5.** The network includes five different PCs, or nodes. The example GenBroker configuration shown here is for PC1, which communicates with PC2, PC3, PC4, and PC5 according to the settings in the GenBroker Configurator. This example demonstrates how communication

between PC1 and the other four PCs can be routed through a mediator node, as shown in Figure 5.5. PC1 will not communicate directly with the remote computers, but it instead will establish a connection with the mediator computer. This mediator will then route the communications to the final destinations—in this case PC2, PC3, PC4, or PC5. The mediator node may be located on a remote computer, which can only be reached through a RAS (remote access service) connection or the Internet.

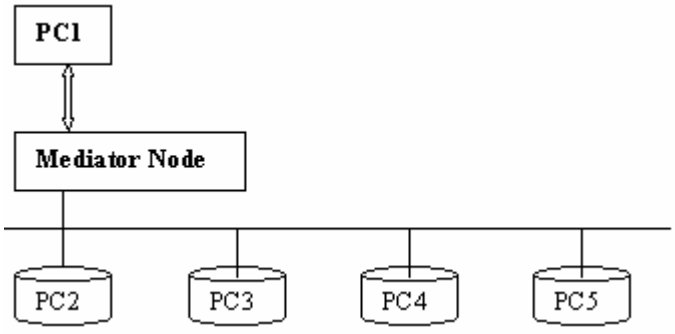


Figure 5.5. Communication via Mediator Node

Communication Using a Mediator Node

The GenBroker configuration on the mediator node determines how the mediator will connect to PC2, PC3, PC4, and PC5. This may use OPC Direct, OPC over DCOM, or OPC over TCP/IP. To configure communication through the mediator node on a TCP/IP channel, for example, do the following:

1. Click the **Add TCP Channel** button. This opens the **Select a Node** dialog box. Specify a name for the mediator node (for example, "Mediator") in the **Node Name or IP Address** field, and then click **OK.** The new channel will appear in the GenBroker Configurator screen as **OPC over TCP/IP Channel via Mediator,** and the new mediator node will appear below the new channel, as shown in **Figure 5.6.**

2. Now add PC2, PC3, PC4, and PC5 to the new TCP/IP channel. Right-click **OPC over TCP/IP Channel via Mediator** and select **Add Node** from the pop-up menu. This opens the **Select a Node** dialog box. Enter **PC2** in the **Node Name or IP Address** field, and then click **OK.** The node name will appear below the channel, as shown in Figure 5.6. Repeat this procedure for PC3, PC4, and PC5.

Note: For additional examples of GenBroker configurations, please see the GenBroker Help documentation by selecting **Help Topics** from the **Help** menu.

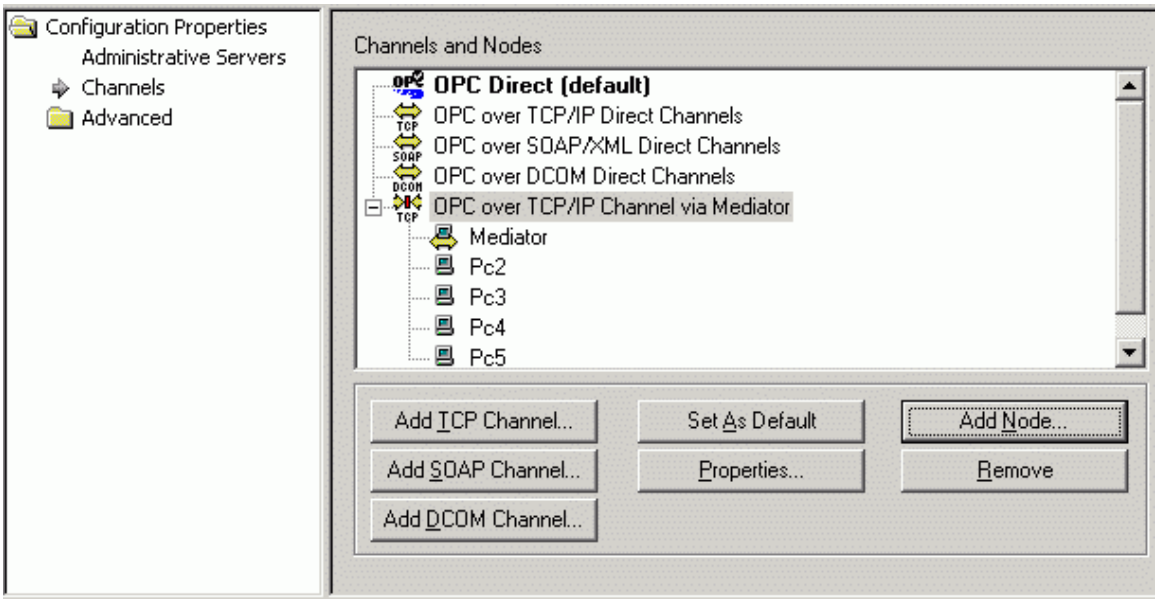


Figure 5.6. Configuring a Mediator Node

Chapter 6

WEBHMI SECURITY

WebHMI Security ActiveX

An example of WebHMI Security is included in the WebHMI sample screens. To access the samples, select **Start > Programs > Smar WebHMI > WebHMI Examples.** The **WebHMI Samples Home Page** appears, as shown in **Figure 6.1.**

smar
WebHMI
Smar' WebHMI enables users to view real-time HMI or SCADA applications over the Internet or intranets on any client PC running Microsoft Internet Explorer or Netscape Navigator. WebHMI is extremely easy to use and set up.
Advanced Web publishing tools allow graphics, trends, alarms and reports to be published to designated Web servers running Microsoft Internet Information Server (IIS), Microsoft Personal Web Server, or Apache Server providing a complete Web-based HMI or SCADA solution.
"Our Web-enabled HMI and SCADA products are based on OPC-To-The-Core technology and offer unparalleled performance, efficiency, ease of connectivity and freedom of choice!"
Default Samples
Industry Samples
Advanced Samples
text inputbox
change color
Hide rectangle
read value
English

Figure 6.1. WebHMI Samples Home Page

Click on **Default Samples** to open the WebHMI **Default Samples** page, as shown in **Figure 6.2.**

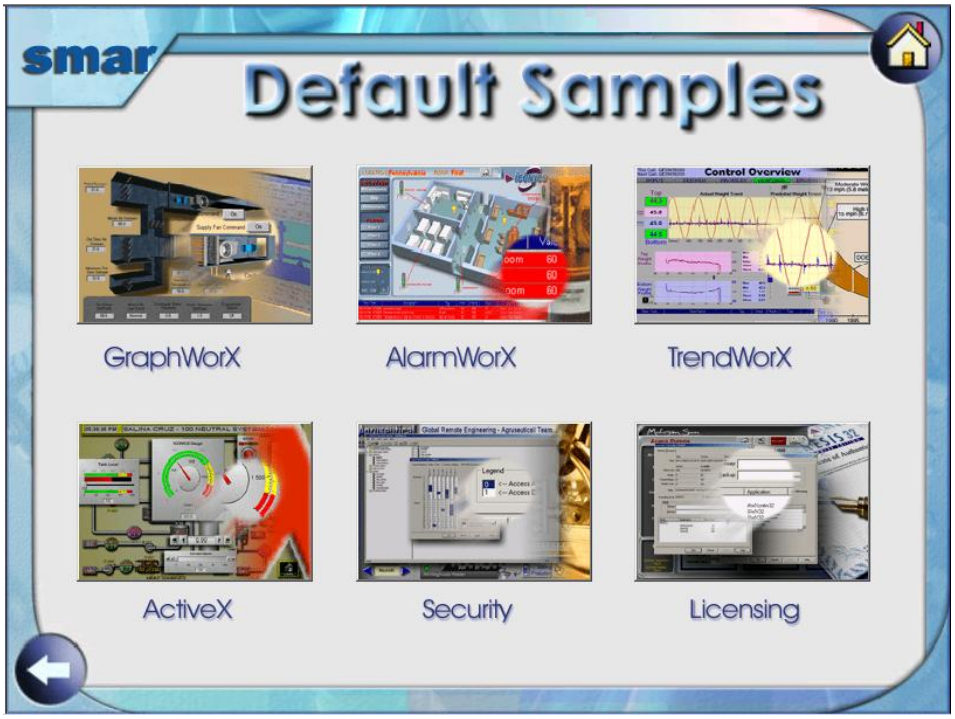


Figure 6.2. WebHMI Default Samples Page

Click on the **Security** button to open the Security sample page shown in **Figure 6.3.** The **Symbol Library** in GraphWorX contains a symbol category file called "WebHMI Security Login.sdf," which contains several symbols that, when dragged into a GraphWorX display, allow users to gain access to the Security Server.

All the symbols use VBScript to call the Security Server on the remote WebHMI Server and get back security information. You do not need to know VBScript to use this symbol. You can directly drag and drop the symbol that you need from the Symbol Library into your display, but you also have the freedom to use the Script Editor toolbar in GraphWorX to change the source code associated with each of these symbols. Or you can copy the code and attach it to your own symbols.

All of the scripts associated with these symbols create an instance of the "Smar Login ActiveX" and call methods of this object or access properties. The complete automation for the "Smar Login ActiveX" is described later in this chapter.
These symbols are shown in Figure 6.3.

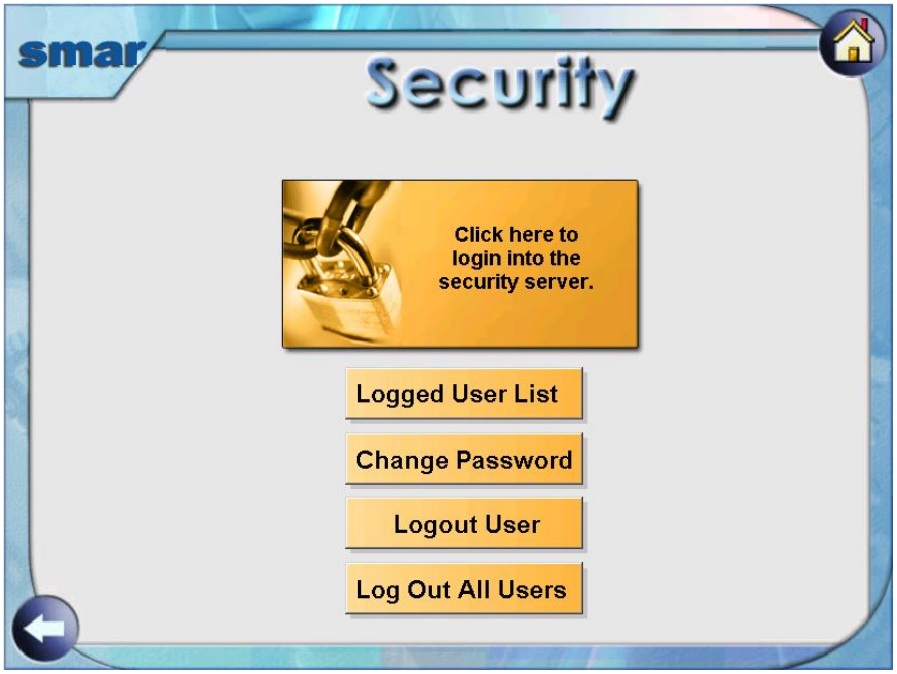


Figure 6.3. WebHMI Security ActiveX Symbols in GraphWorX

Logging Into the Security Server

The **WebHMI Security Login ActiveX** symbol button, shown in **Figure 6.4,** enables WebHMI users on remote client machines to log in to the Security Server. For example, if the "Login" symbol button is placed in a GraphWorX display, the user can simply click on the symbol (button) in runtime mode to launch the **Security Login** dialog box, as shown in Figure 6.4.

Figure 6.4. Security Login Symbol Button

The **Security Login** dialog is basically the same as the one for the Security Login Utility, except that the **Advanced** login mode is disabled, as shown in the **Figure 6.5.** The WebHMI Security Login ActiveX also includes full keypad support (ideal for touch-screen systems). The Login ActiveX allows simultaneous login of many users (this must be enabled on the Security Server global settings).

The drop-down list for the user name can show:

- The complete list of users in the system.
- The list of the currently logged users.
- The name of the last logged user.

All of these features must be enabled on the Security Server in order to work.
For more information, please see the Security Configurator Help documentation.

Note: When you log into the Security Server using the Login ActiveX, you do not get any warning messages when the security session is about to expire. If your security session expires, then the Login ActiveX will automatically be displayed again.

Figure 6.5. Logging Into the Security Server

Changing the Security Server Password

If you should wish to change your password you can do it by clicking on the **Change Password** button on the login ActiveX dialog, or you can do it directly by using the **Change Password** symbol button and dragging it into your GraphWorX display, as shown in **Figure 6.6.**

Figure 6.6. Change Password Button

Clicking the **Change Password** button opens the **Security Password Change** dialog box, as shown in **Figure 6.7.** Type your new password in the **New Password** and **Retype Password** fields, and then click the **OK** button.

Figure 6.7. Changing the Security Password

Viewing the Logged User List

To view a list of users currently logged in to the Security Server, click the **Logged User List** symbol button in your GraphWorX display, as shown in **Figure 6.8.**

Figure 6.8. Logged Users List Button

The **Security** window will appear, as shown in **Figure 6.9.** The **Security** window allows you to view the list of users that have logged in.

Figure 6.9. List of Logged Users

Logging out of the Security Server

To log out everyone who has logged in, you can use the **Log Out All Users** symbol button shown in **Figure 6.10.**

Figure 6.10. Logout All Button

You can also logout one specific user with a simple click on the **Logout User** button, as shown in **Figure 6.11.**

Figure 6.11. Logout User Button

Note: You have to specify the user to be logged out in the VBScript code associated with this button. You can do it by editing the script with the Script Editor toolbar in GraphWorX.

Security OLE Automation

The OLE Automation interface for the WebHMI Security ActiveX is compatible with VBA and VBScript. You can perform login/logout operations directly trough scripting without displaying any user interface. The WebHMI Security ActiveX contains the following OLE Automation interfaces:

LoginDlg()
Launches the login dialog.

ChangePwdDlg()
Launches the dialog to change the password.

ShowLoggedInUsers()
Launches the dialog to show a list of the usesr currently logged into the Security Server.

Logout()
Logs out all currently logged users.

SetTimeout(LONG nSec)
Sets the timeout for all of the GenClient calls to the Security Server.

ShowResultMsgs(BOOL bShow)
Enables / disables the message box with the result (e.g. "failed to log on to the Security Server").

LoginUser(BSTR username, BSTR password)
Logs in one specific user through code.

LogoutUser (BSTR username)
Logs out a specific user through code.

GetLoggedInUsersNames (BSTR usernames)
Gets the list of currently logged users. The string "usernames" is filled with the comma-separated list of currently logged user names. Note that by default the Security Server does not allow concurrent login of multiple users; the concurrent login option must be enabled from the Security Server Configurator. Please refer to the Security Server documentation for additional details.

Launching the Security Login ActiveX Through Scripting

The Security Login ActiveX can be programmatically created and initialized from VBA Script, VBScript and JScript. The GraphWorX Symbol Library contains a category named "WebHMI Security Login" located under the "VBAScriptSymbol" folder, which is filled with samples of each of the automation interfaces listed in Section 6.6. Please refer to the aforementioned samples for additional information on how to use the Login ActiveX through scripting.

The following code sample has been extracted from the Symbol Library; this sample shows how to launch the Login ActiveX from VBScript. The code runs on WebHMI too:

```
Set t = CreateObject ("Smar.LoginActiveX")
If t Is Nothing Then
 MsgBox "An error has occurred while trying to launch the login dialog."
Else
 t.LoginDlg()
End If
```

The following code sample has been extracted from the Symbol Library; this sample shows how to get the list of currently logged users from VBScript. The code runs on WebHMI too:

```
Set t = CreateObject ("Smar.LoginActiveX")
If t Is Nothing Then
 MsgBox "An error has occurred while trying to launch the login dialog."
Else
   t.GetLoggedInUsersNames str
 MsgBox str
End If
```

WEBHMI LICENSING

Viewing Licensing Information Using WebHMI

An example of WebHMI Licensing is included in the WebHMI Sample screens. To access the samples, select **Start > Programs > Smar WebHMI > WebHMI Examples.** The **WebHMI Samples Home Page** will appear, as shown in **Figure 7.1.**

smar
WebHMI
Smar' WebHMI enables users to view real-time HMI or SCADA applications over the Internet or intranets on any client PC running Microsoft Internet Explorer or Netscape Navigator. WebHMI is extremely easy to use and set up.
Advanced Web publishing tools allow graphics, trends, alarms and reports to be published to designated Web servers running Microsoft Internet Information Server (IIS), Microsoft Personal Web Server, or Apache Server providing a complete Web-based HMI or SCADA solution.
"Our Web-enabled HMI and SCADA products are based on OPC-To-The-Core technology and offer unparalleled performance, efficiency, ease of connectivity and freedom of choice!"
Default Samples
Industry Samples
Advanced Samples
text inputbox
change color
Hide rectangle
read value
English

Figure 7.1. WebHMI Samples Home Page

Click on **Default Samples** to open the WebHMI **Default Samples** page, as shown in **Figure 7.2.**

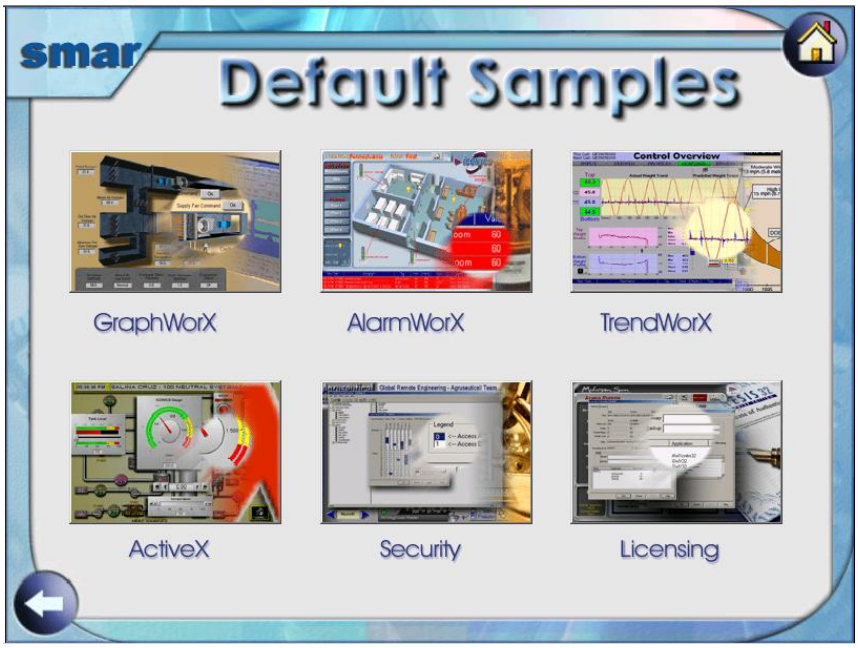


Figure 7.2. WebHMI Default Samples Page

Clicking on the **Licensing** button opens the **Licensing** page, which allows you to launch the ProcessView License Monitor. The License Monitor allows you to get an immediate view of the number of Client Units your enterprise is using at a given time. Click the **Show License** button, as shown in **Figure 7.3.**

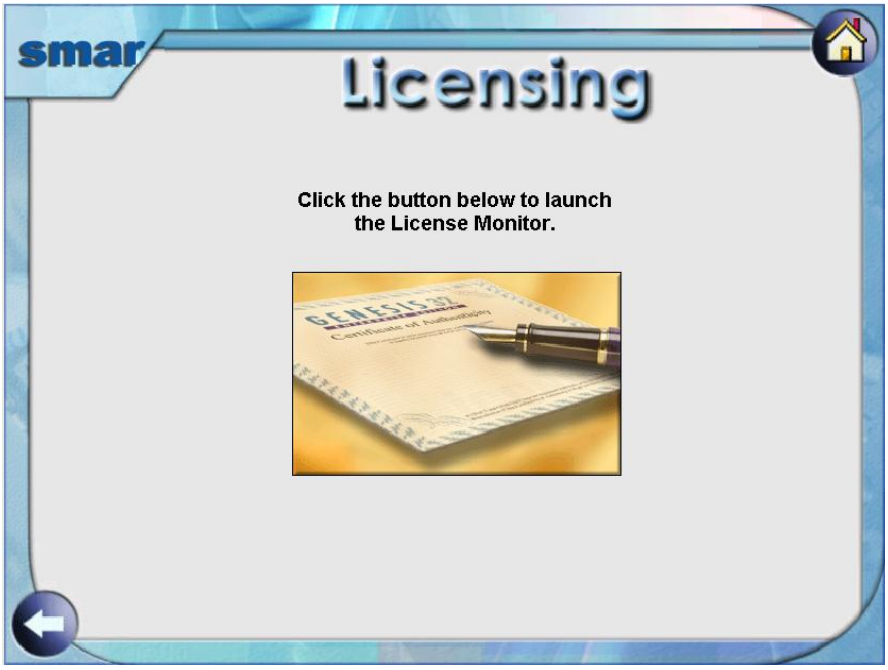


Figure 7.3. Launching the License Monitor

This launches the **License Monitor,** shown in **Figure 7.4,** which provides information about the Client Units and points used to run specific ProcessView applications. The Client Units field lists the number of Client Units that are currently being used in all ProcessView applications, the number of **Available** Client Units (which are not currently in use), and the **Total** number of Client Units, which is equal to the number of available Client Units plus the number of Client Units In Use. For example, if you have 700 total Client Units and 300 Clients Units are currently being used, then 400 Client Units are available. Similarly, the Points field provides information about the number of process points that are currently being used (in runtime mode) in all ProcessView applications.

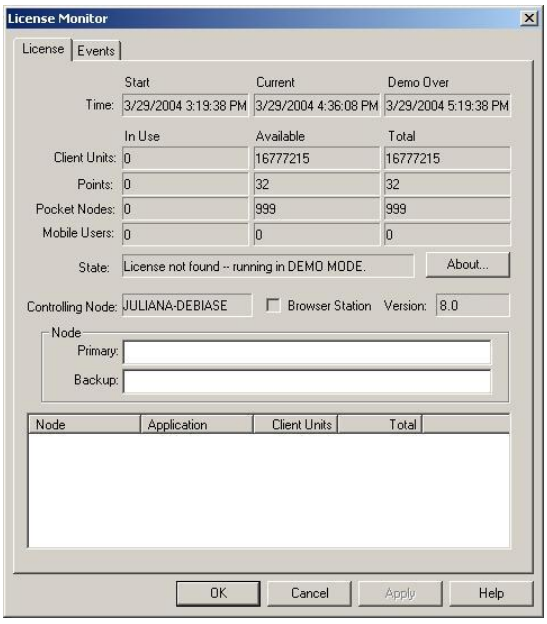


Figure 7.4. ProcessView License Monitor

License for the WebHMI Server

As mentioned in Chapter 2, before installing WebHMI you must have a licensed ProcessView installed on the designated WebHMI server. Likewise, the Smar Software License Utility must also be installed on the WebHMI server.

Otherwise, the Software License Utility and its installation are the same for WebHMI as they are for any other ProcessView application.

Obtaining Additional Connection Units

If you desire additional units, you will need additional licensing for the WebHMI server. Additional units are provided in increments of five. To obtain additional units, contact your local Smar distributor.

Licensing OLE Automation

The OLE Automation interface for the WebHMI License ActiveX is compatible with VBA and VBScript. You can use the OLE automation to launch the License Monitor and review license information:

ShowLicense()

Call this method to show the license monitor.

Launching the License Monitor ActiveX Through Scripting

The License Monitor ActiveX can be programmatically created and initialized from VBA Script, VBScript and JScript. The GraphWorX Symbol Library contains a category named “WebHMI Security Login” located under the “VBAScriptSymbol” folder, which contains samples of the automation interface listed in Section 7.4. Please refer to the aforementioned sample for additional information on how to use the License Monitor ActiveX through scripting.

The following code sample has been extracted from the Symbol Library; this sample shows how to launch the License Monitor ActiveX from VBScript. The code runs on WebHMI too:

```
Set t = CreateObject ("Smar.LicenceMonitorActiveX.1")
If t Is Nothing Then
 MsgBox "An error has occurred while trying to launch the login dialog."
Else
 t.ShowLicense()
End If
```

Chapter 8

CHANGE HOST UTILITY

Introduction to the Change Host Utility

At some point, you might find that you wish to switch Web servers or use an OPC server on a different node. The WebHMI Change Host Name Utility enables you to do this easily. To open the Change Host Utility from the Windows **Start** menu, select **Programs > Smar WebHMI > Change Host Name Utility.** The **Change Host Name** dialog box will appear, as shown in **Figure 8.1.**

Note: The process of running the Change Host Utility may require several minutes, and during this time the circle associated with each file type shown in Figure 8.1 (on the right side) will change color. The circle will be red while processing a specific file type, and it will be green when the specific file type is completed with success. The screen may actually flash for an instant while the utility is "operating" on a file. It is possible to see the file currently processed by reading the file name in the bottom right part of the dialog.

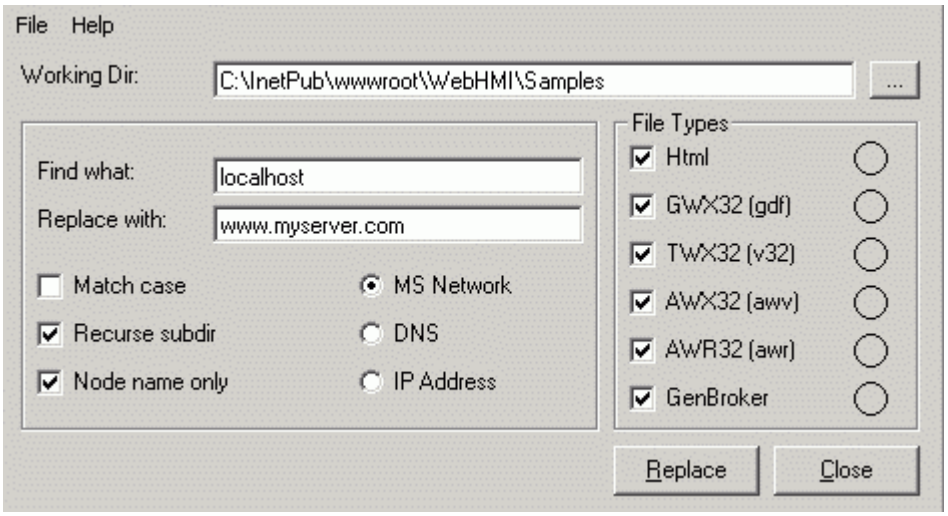


Figure 8.1. Change Host Name Dialog Box

Changing the Directory Name

Change the directory name by typing it in the **Working Directory** field, as shown in **Figure 8.2.** You can also change the directory name by clicking on the button to the right of the **Working Directory** field or by selecting **Set Working Directory** from the **File** menu.

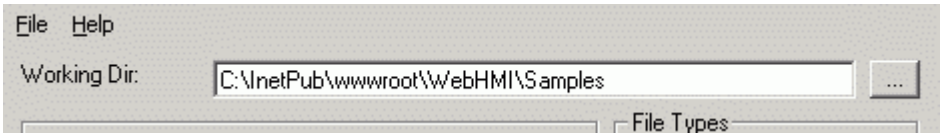


Figure 8.2. Directory Name

Finding and Replacing Files

Functioning like any "find-and-replace" command, the **Change Host Name** dialog box searches ProcessView files for:

- **GWX32:** GraphWorX (.gdf) display files
- **TWX32:** TrendWorX (.v32) files
- **AWX32:** AlarmWorX Viewer (.awv) files
- **AWR32:** AlarmWorX Report (.awr) files
- **GenBroker:** GenBroker configuration (.gbx) files
- **HTML:** .htm files

Choose the type of file for which you want to search by clicking on the check boxes in the right column, as shown in Figure 8.1. The **Change Host Name** utility searches for the text typed in the **Find what** field and replaces it with the text typed in the **Replace with** field, as shown in **Figure 8.3.**

Figure 8.3. Find What and Replace With Fields

Check Boxes
Check the **Match case** check box to do case-sensitive searches, and check the **Recurse Subdirectory** check box to search all subdirectories of the working directory. A close-up view of this section of the dialog box is shown in **Figure 8.4.**

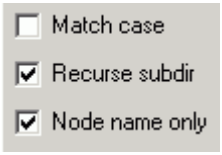


Figure 8.4. Check Boxes

Node name only is a special check box that uses **Find String** and **Replace String** and adds "\\" or "//" at the beginning and "\" or "/" at the end of these strings. This confirms that you have really replaced network node names.

You can modify the AlarmWorX files and GenBroker Configuration files only by selecting **Node name only.** There is no other text information that you can change. The check boxes for the AlarmWorX and GenBroker Configuration files will be grayed out if you uncheck the **Node name only** check box.

Thus, when you uncheck **Node name only,** you can use the Change Host Name utility to change, for example, a point name in GraphWorX displays and TrendWorX files at the same time. You must then enter the original name in the **Find What** field and the new name in the **Replace with** field.

Other Options
The three options to the right of the check boxes allow you to change the name or address in the **Replace with** field. As shown in **Figure 8.5,** a name appears in the **Replace with** field if you select the **MS Network** option.

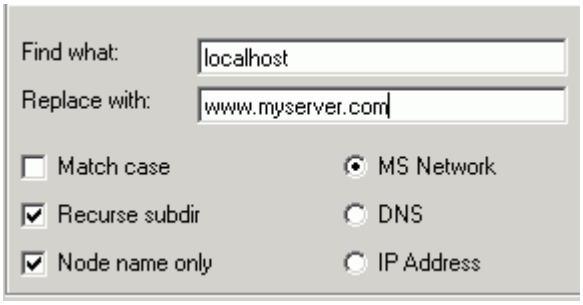


Figure 8.5. MS Network Option

Similarly, another name or address will appear in the **Replace with** field if you select the **DNS** or **IP Address** option, as shown in **Figure 8.6** and **Figure 8.7,** respectively. Click the **Close** button to exit from the Change Host Name Utility.

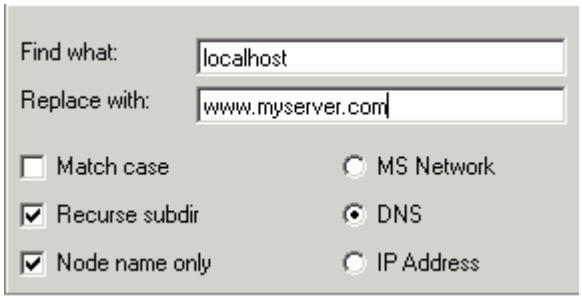


Figure 8.6. Domain Name Services Option

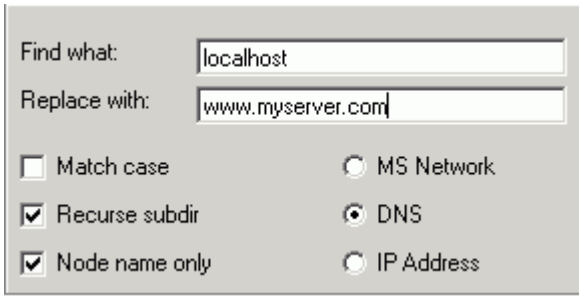


Figure 8.7. IP Address Option

Chapter 9

WEBHMI VERSION SUPPORT

Introduction to Version Support

WebHMI is a thin-client application that uses ActiveX technology to deliver graphics, alarms and trends from the server machine to client machines. WebHMI supports thin clients, which means that WebHMI clients do not need to have any software installed.

When ActiveX components are downloaded from a WebHMI server to a WebHMI client over the Internet, the ActiveX components are silently installed on the client machine. The ActiveX components are delivered into the .cab files archive. The .cab files are referenced in the HTML code near the corresponding ActiveX "clsid". Microsoft Internet Explorer can detect whether ActiveX components are present on the client machine. Thus, if an ActiveX is already installed on the client machine, then there is no need to download a 2 MB .cab file for that ActiveX. In other words the cab files are downloaded only the first time that the WebHMI client station connects to the WebHMI server station. Once the cab files are downloaded, they will not be downloaded any more.

Under certain circumstances, you may want to upgrade your WebHMI server with a newer version of the cab files, for example when upgrading to a new major release of WebHMI, or when installing a service pack.

Upgrading WebHMI is a two-step process:

1. You need to upgrade the cab files located on the server machine.
2. You need to download the new cab files to the client machine(s).

Step 1 is reduced to a simple “file copy” operation. Step 2 requires the use of the WebHMI Update Versions Utility.

The Update Versions Utility processes the files installed on the server, reviews the version information contained inside them, and upgrades it to the latest version of components installed on the WebHMI server.

Once the version information is updated, the client machine will recognize that the server contains a newer version of the components used for visualization, alarming and trending, and will automatically re-download the new cab files with the new components.

For example, suppose that a client machine is running WebHMI Version 8.0, and you want to connect the client machine to a WebHMI Version 8.1 server. In this case, the client machine requires GenBroker Version 8.1 support in order to communicate with the WebHMI 8.1 server. It also requires GraphWorX, AlarmWorX, TrendWorX, and all other ActiveX components from Version 8.1. The ProcessView ActiveX components are backward compatible; for example, you can open a Version 7.0 file from a Version 8.0 ActiveX component, but you cannot open a Version 8.0 sample file if you have a Version 7.0 ActiveX component.

Now, for example, suppose that the client has already been connected to a WebHMI 8.0 server, and the client has already installed all the ActiveX support for ProcessView Version 8.0. Also suppose that you want to connect the same client to a WebHMI 8.1 server. If there is no version information in an HTML page, Internet Explorer will not download the new ActiveX .cab files because the ActiveX with the same ID is already registered on the client machine.

To avoid such compatibility problems, WebHMI adds the correct version number information to HTML pages. This is accomplished with the **Update Versions Utility.** This utility gets the version information from an information (.inf) file called "WebHMI.inf," which contains information about the entire WebHMI installation, including which archives are installed in which directories, what component they contain, and the version numbers of the components installed.

Update Versions Utility

The WebHMI Update Versions Utility manages all versioning information on the server side. The utility allows you to modify HTML files by adding version information for ActiveX components. The version information in HTML files is necessary for the automatic update of ActiveX components. If an older version of an ActiveX is already installed and registered on your machine, a newer version will be downloaded, but only if the version information is placed in the HTML file.

The version information is taken from the "WebHMI.inf" file mentioned above. In WebHMI Version 8.0, the "WebHMI.inf" file is installed by default on the WebHMI server machine in the "C:\InetPub\wwwroot\WebHMI" directory.

The Versions Utility is similar to other "find-and-replace" utilities that search in a specified directory for entered files. The utility searches for the occurrence of ActiveX objects in HTML files. If it finds any, it then searches the "WebHMI.inf" file for the updated version and inserts the new version information into the HTML files.

Starting the Versions Utility

To open the Versions Utility from the Windows **Start** menu, select **Programs > Smar WebHMI > Versions Utility,** as shown in **Figure 9.1.**

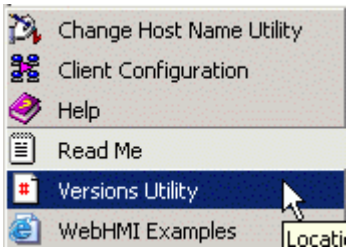


Figure 9.1. Starting the Versions Utility

The **Update Versions** dialog box appears, as shown in **Figure 9.2.**

Note: The Versions Utility is Web-server oriented and should be used by the Web server (WebHMI server) administrator only.

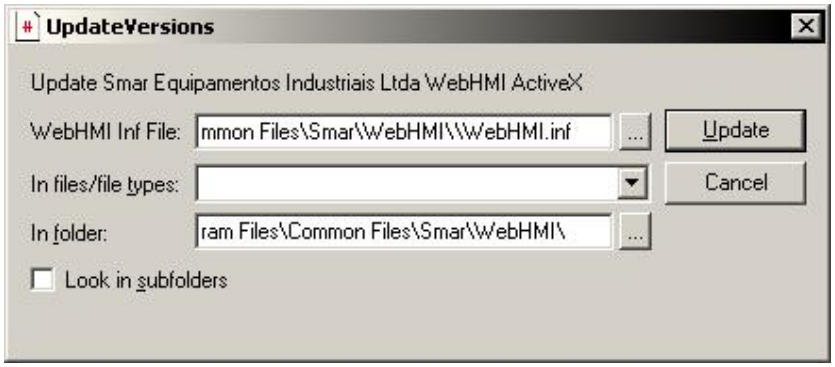


Figure 9.2. Update Versions Dialog Box

The **WebHMI Inf File** field specifies a path to the information file, which contains version information for ActiveX components. The default file is the "WebHMI.inf" file described above. If you want to change the file or its location, you can simply type the new location, or click the **...** button to browse for the file.

The **In Files/File Types** field contains a list of files and/or file types to search for. The default file types are *.htm and *.html; the utility will search for all .htm and .html files in the specified location.

The **In Folder** field contains the name of the folder where the search will be started. You can also use the **...** button to change the folder. If the **Look In Subfolders** check box is checked, the search will be recursive for all subdirectories of the selected folder.

When you click **Update,** the utility will search for the specified files in the selected directories. If the utility finds any of them, it will open each file and search the file for the occurrence of an ActiveX. If an ActiveX is found, the utility will search for version information for that ActiveX in the "WebHMI.inf" file.

If the information is found, the updated version information will be automatically appended to the .cab file address in the HTML code. For example, a GraphWorX Viewer ActiveX (GWXView32.cab) with version 8.00.137.0 in a Web page will be appended to the "CODEBASE" attribute of the "OBJECT" block in the HTML code as follows:

CODEBASE=http://www.WebHMI.com/WebHMI/cabs/**GWXview32.cab#Version=8.00.137.0**

Appendix A

ALARMWORX AND TRENDWORX WEB ACCESS

Introduction

WebHMI includes AlarmWorX Web Access support for reports and operator comments. The configuration interfaces for the AlarmWorX Report ActiveX and the AlarmWorX Viewer ActiveX enable remote access to Alarm Logger databases over the Internet.

Web Access for Alarm Reports

The AlarmWorX Report ActiveX enables clients to download alarm reports over the Internet. The **Web Access** tab of the **Alarm Report ActiveX Properties** dialog box, shown below, allows you to access the currently connected Alarm Logger database over the Internet via a **Remote Data Service (RDS), or Remote Database Agent (RDA).** The RDS, which is hosted by an Internet Information Server (IIS), enables the downloading of report data from the logger database to a client over the Internet. Simply specify the URL name or IP address of the primary IIS Web server in the **Primary Internet Information Server for RDA** field, as shown in the figure below. The **Remote Recordset Page Size** field specifies the size of the recordset portion in the database records.

Note: You also may designate a redundant IIS Web server by specifying the URL name or IP address of the redundant IIS Web server in the **Backup Internet Information Server for RDA** field, as shown **Figure A.1.**

Note: For more information, please see the AlarmWorX Report ActiveX Help documentation.

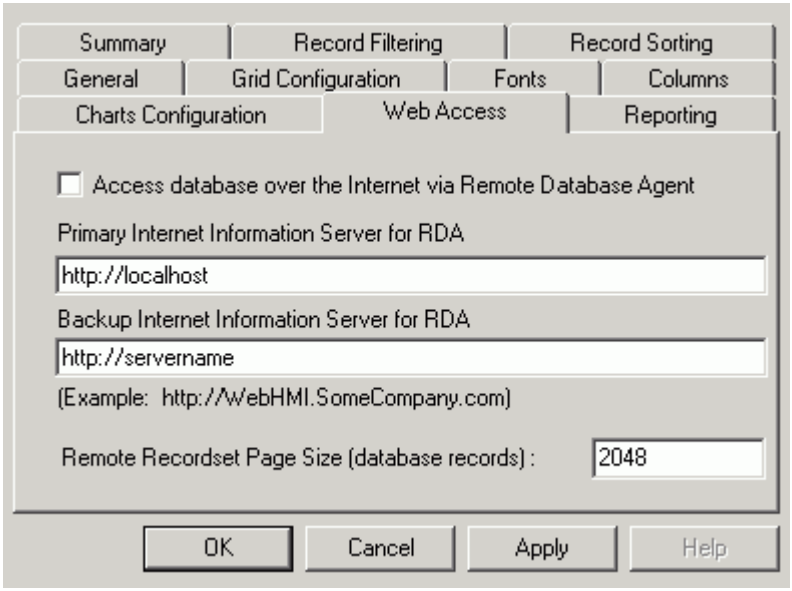


Figure A.1. Alarm Report ActiveX Web Access Tab

When Internet Access to the logger database is enabled, the client makes a request over the Internet to the IIS Remote Data Service. The RDS uses a special component, called the Data Factory Handler, to relay the request to the local logger database on the server, as shown in **Figure A.2.** The RDS retrieves the relevant information for the alarm report from the logger database so that the client can download the data from the server over the Internet. The client machine then displays the data in the Report ActiveX per the client's report configuration settings.

In most typical scenarios, the Report ActiveX configuration is done on the server side. Those steps include:

- Connection to the database is established by clicking the **Connection Parameters** button on the **General** tab of the **Report ActiveX Properties** dialog and using the **Database Connection** dialog.
- Internet access settings are enabled on the **Web Access** tab.
- Configuration data saved as an *.awr file or a *.gdf display containing the Report ActiveX are published to an HTML document on a WebHMI server.

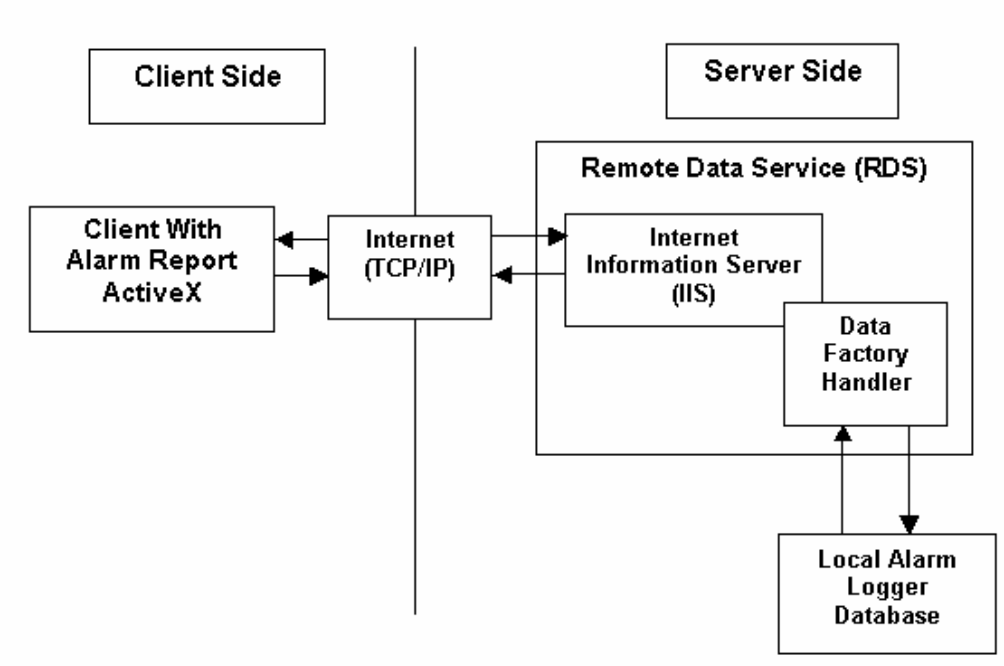


Figure A.2. Downloading Alarm Report Data From a Remote Logger Database

Web Access Support for Alarm Viewer Operator Comments

The AlarmWorX Viewer ActiveX enables clients to download operator comments from a remote database over the Internet. The **Display** tab of the **Alarm Viewer ActiveX Properties** dialog box, shown in **Figure A.3,** allows you to connect to an operator comments database.

Note: For more information, please see the AlarmWorX Viewer ActiveX Help documentation.

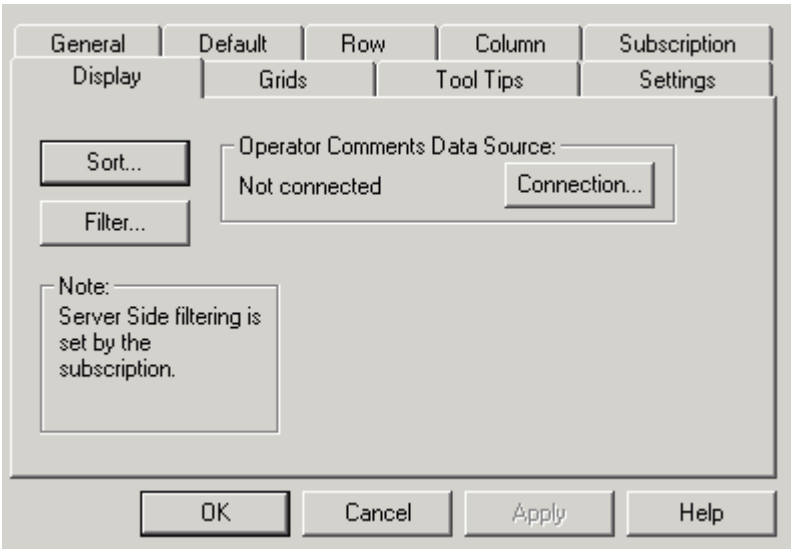


Figure A.3. Alarm Viewer ActiveX Properties: Display Tab

Connecting to the Operator Comments Database

In most situations, you will use the **AWXLog32.mdb** Microsoft Access database (located in the ProcessView Bin directory) as your operator comments database. This is the standard Alarm Logger database. To connect to the operator comments database:

1. Click the **Connection** button on the **Display** tab. This opens the **Database Connection** dialog box, as shown below.

2. Under **Data provider,** select **Microsoft Access,** as shown in **Figure A.4.**

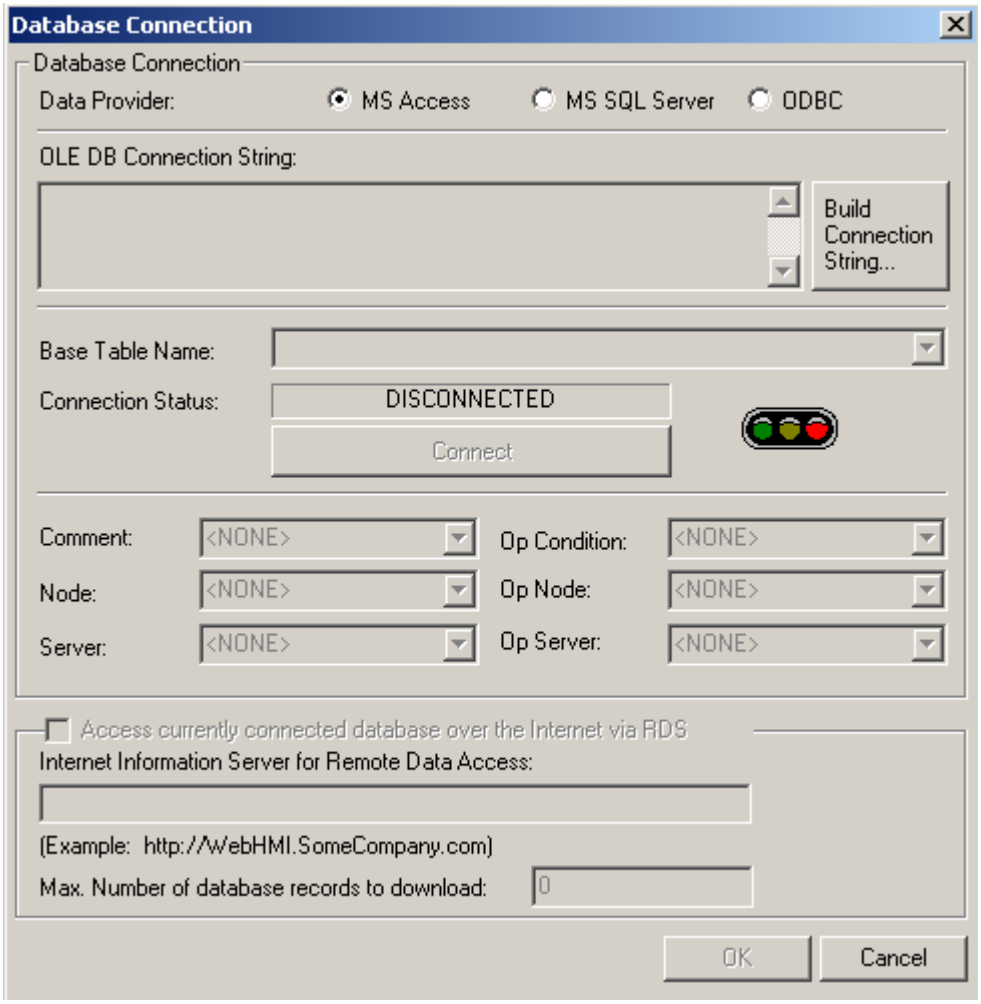


Figure A.4. Connecting to a Database

3. Click the **Build Connection String** button. This opens the Microsoft **Data Link Properties** dialog box, as shown in **Figure A.5.**

Note: All data source connections are made through the **Data Link Properties** dialog box. The **Connection** tab settings may vary depending on which data provider you selected. Click the **OK** button. Click the **Help** button at any time to view the Microsoft Data Link help documentation.

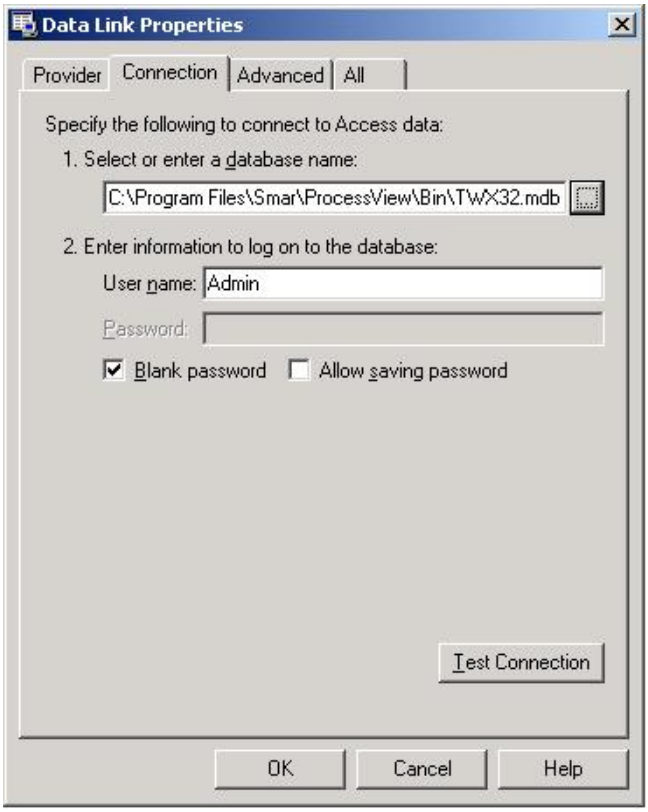


Figure A.5. Specifying a Data Source

4. Click the **…** button and select the **AWXLog32.mdb** file from the bin directory, as shown in **Figure A.6.** Click the **Open** button.

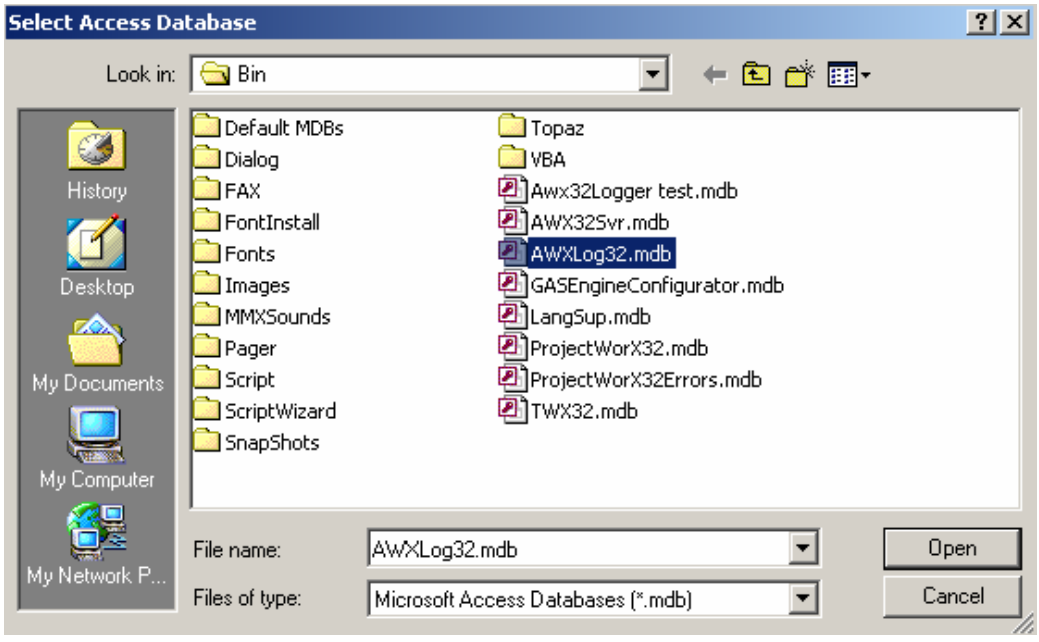


Figure A.6. Selecting the Operator Comments Database

5. The data source reference appears in the **Database Connection** dialog box in the **OLE DB Connection String** field, as shown below. In the **Base Table Name** field, select **EventLog** from the drop-down list.

Note: The **Base Table** list varies according to the active Alarm Logger configuration database.

6. Click the **Connect** button to connect to the database. The traffic light icon changes to green when the connection is successful, as shown in **Figure A.7.**

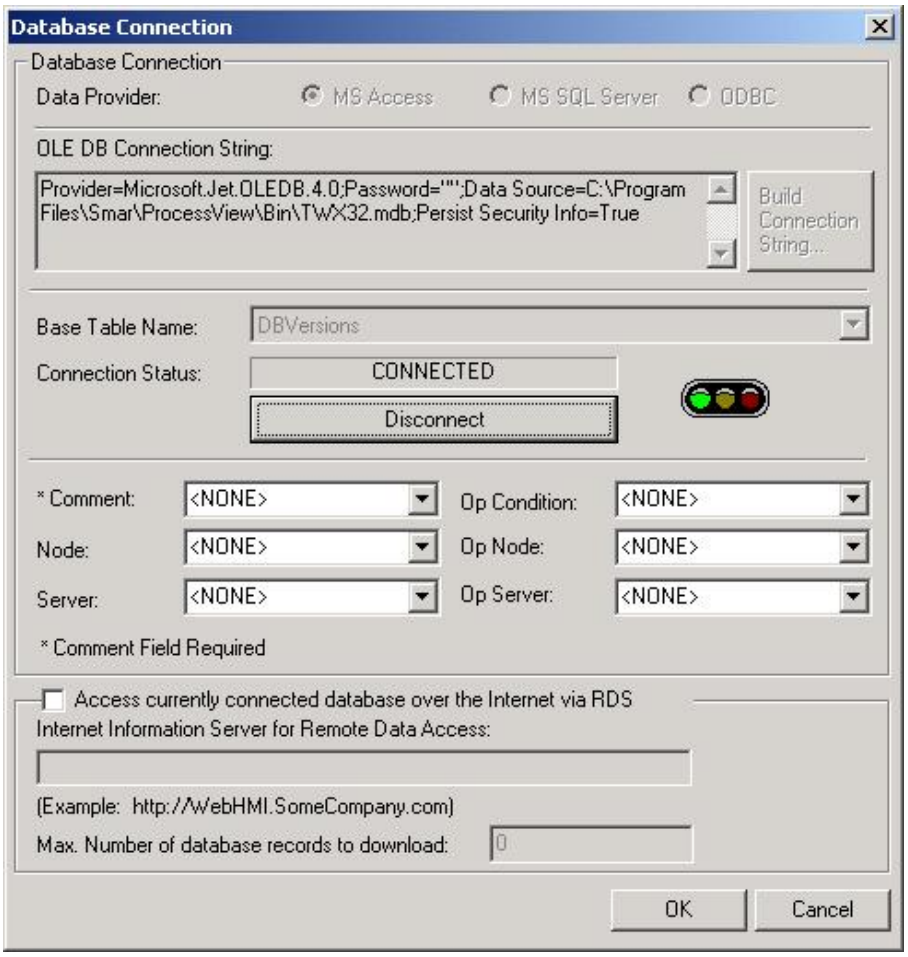


Figure A.7. Database Connection

7. Now you must specify which column in the database table you will use to for logging and storing comments. In the **Comment** box, select a column from the drop-down list. For most cases, only the **Comment** box should be configured. The exception is the case where the system is logging from two different alarm servers, using the same tag names in both alarm server configurations. This is the only case where the **Node, Server, Op Condition, Op Node,** and **Op Server** selections should be used.

Enabling Remote Web Access to Operator Comments

The Alarm Viewer enables you to gain access to operator comments from the currently connected Alarm Logger database over the Internet. To enable Web access to operator comments, check the **Access currently connected database over the Internet via RDS** check box at the bottom of the **Database Connection** dialog box. A **Remote Data Service (RDS),** which is hosted by an Internet Information Server (IIS), enables the downloading of operator comments data from the logger database to a client over the Internet.

Once you have connected to the operator comments database. Simply specify the URL name or IP address of the IIS Web server in the **Internet Information Server for Remote Data Access** field in the **Database Connection** dialog box. You can also specify a maximum number of database records to download from the logger database.

When Internet Access to the logger database is enabled, the client makes a request over the Internet to the IIS Remote Data Service. The RDS uses a special component, called the Data Factory Handler, to relay the request to the local logger database on the server, as shown in **Figure A.8.**The RDS retrieves the relevant information for the alarm report from the logger database so that the client can download the data from the server over the Internet. The client machine then displays the data in the Viewer ActiveX per the client's configuration settings.

In most typical scenarios, the Viewer ActiveX configuration is done on the server side. Configuration

data saved as a *.awv file or a *.gdf display containing the Viewer ActiveX are published to an HTML document on a WebHMI server.

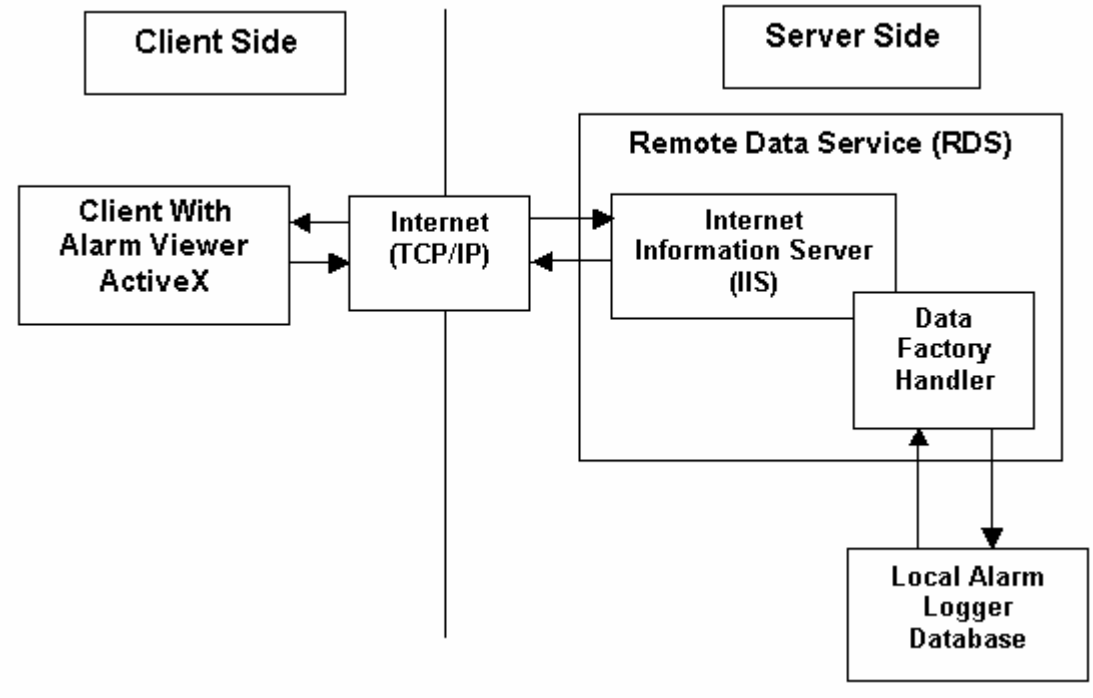


Figure A.8. Downloading Operator Comments From a Remote Logger Database

Tips for AlarmWorX Remote Data Access

A good understanding of Remote Data Access principles is a key element to building a successful solution for a Web-enabled alarm reporting system. Since the volume of alarm databases can reach hundreds of megabytes, it is very important to keep in mind the following aspects.

Maximum Number of Downloadable Database Records and Adequate Record Filtering
The setting for the maximum number of records to download (Nmax) indicates that any database query operations would return no more than Nmax alarm records (lines to the grid). If more than Nmax records from the current table match the current filtering criteria, then only the portion of the record set containing top Nmax records is returned. To avoid such record set cut-off, strong restrictive record filters should be used all the time. Since the record filters are processed on the database side, the database server uses all existing records to build the record set.

Proper Choice of the Back-end Database
It should be noted that the use of Microsoft Access databases (*.mdb) is not recommended for Web Access. Poor memory management in Microsoft Jet Engine creates a serious scalability problem for any type of Web database application. **The recommended back-end database for Web alarming is MSDE (Microsoft Desktop Engine) or Microsoft SQL Server.**

Table Management in Alarm Logger Configuration
To provide further optimization for the Alarm Logger database operations, the table management should be enabled in the logger configuration. See the Alarm Logger Configurator documentation for more information.

Using the Remote Database Access Manager
The **AlarmWorX Remote Database Access Manager,** shown in **Figure A.9,** is a separate utility (the **AWXRep32RDSSM.exe** file) located in the ProcessView Bin directory. This utility configures the communication parameters for the Data Factory Handler (described above) so that the AlarmWorX Report ActiveX can connect to a remote Alarm Logger database and retrieve data from the database. It also enables security for remote database access via the ProcessView Security Server.

Figure A.9. AlarmWorX Remote Database Access Manager

Database Connection Options

The **Database Connection Options** section of the AlarmWorX Remote Database Access Manager sets the following connection parameters for the Data Factory Handler:

- **Connection Retry Number:** Specifies the maximum number of times the Data Factory Handler will try to reconnect to the Alarm Logger database in case the connection fails.
- **Connection Retry Delay:** Sets the amount of time (in seconds) the Data Factory Handler waits between subsequent connection retries to the Alarm Logger database.

Security Management

The AlarmWorX Remote Database Access Manager allows you to enable security for remote database access via the ProcessView Security Server. Check the **Enable Security Management for Remote Database Access (via Security Server)** check box. When security is enabled, each time a user tries to access the Alarm Logger Database, the Security Server will check to verify whether that user is allowed to have remote access to the database.

Note: For information about the Security Server configuration and login, please see the Security Configurator and Security Login Help documentation.

Because the database is accessed over the Internet via TCP/IP communications, GenBroker communications over OPC over TCP/IP must also be enabled. Thus, you must specify a **GenBroker Configuration File** (.bgx). Clicking the **…** button to the right of this field allows you to browse for a GenBroker configuration file, as shown in **Figure A.10.**If you do not specify a GenBroker configuration file, or if no Security Server is specified in the GenBroker configuration, then the default system GenBroker configuration file will be used.

Note: If you are using WebHMI, it is recommended that the Security Server be located on the same machine as the WebHMI Server. For information about how to configure GenBroker communications, please see the GenBroker Help documentation.

The **GenClient Updates Timeout** sets the maximum amount of time (in seconds) the Data Factory Handler will wait for Security Server validation of a data request from the client (i.e. the client with the Alarm Report ActiveX installed) before returning an "Access Denied" error.

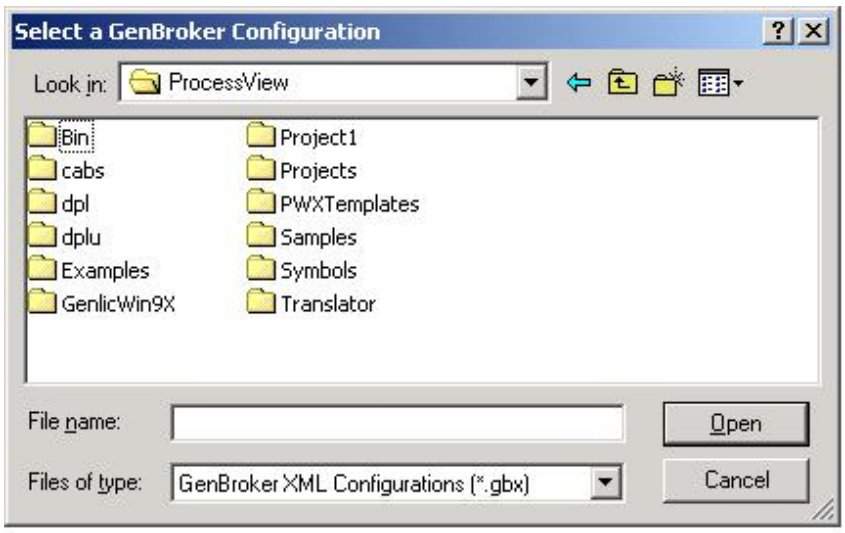


Figure A.10. Selecting a GenBroker Configuration File

TrendWorX Reporting Tool

To open the TrendWorX Reporting tool from the Windows Programs menu, select Programs > Smar ProcessView > ProjectWorX > TrendWorX > TrendWorX Reporting.

The **TrendWorX Reporting** screen will appear, as shown in **Figure A.11.**

Figure A.11. TrendWorX Reporting Screen

The TrendWorX Reporting tool exports .csv files, databases, and MS Excel Workbooks. Of these, only Excel Workbooks can be imported into HTML files. To do this, configure an Excel-based report by selecting **Add Report** from the **Edit** menu. This opens the **Report Configuration Wizard,** which is a series of dialog boxes that allow you to configure report. In the **Target** tab, shown in **Figure 4.12,** select **Excel File,** check the **Publish** option, and click on the **HTML Path** button to determine where in the directory you wish to place the files. Currently, some server-side scripting may be required to generate dynamically HTML pages with all the available reports. In future versions of WebHMI, the WebHMI Explorer based on MS Digital Dashboard will do this automatically.

For a complete guide through the steps in using the TrendWorX Reporting Tool, please see the TrendWorX user guide or online help documentation.

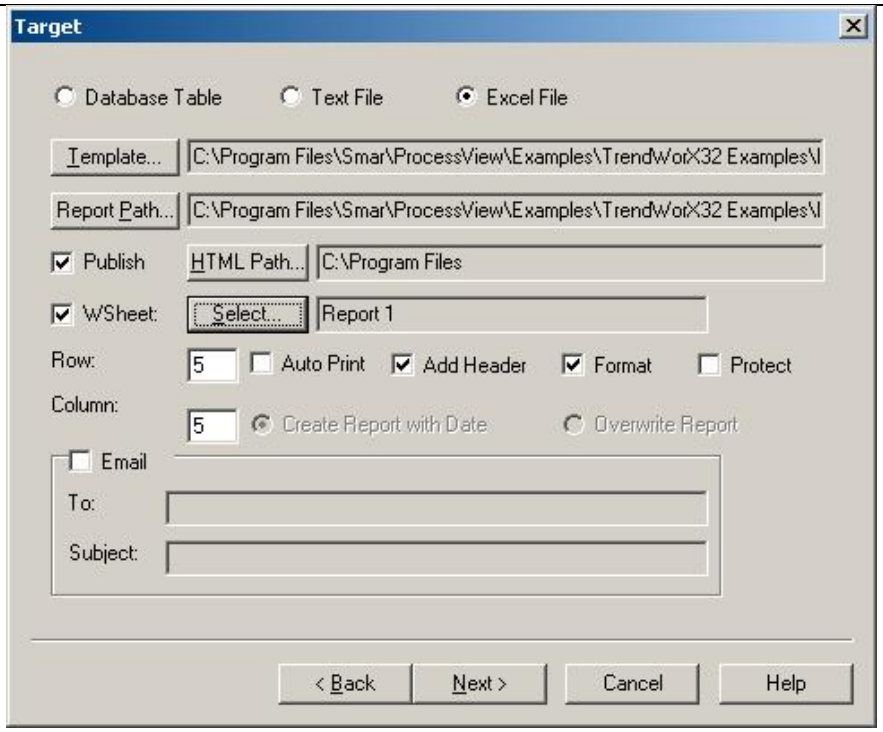


Figure A.12. TrendWorX Report Configuration Wizard

Appendix B

SETTING UP DCOM

DCOM As a Requirement

DCOM is now essential to an up-to-date Windows environment. DCOM is built into Windows NT. DCOM settings must be configured carefully and correctly to be able to view WebHMI on the server.

The Web server should also have ProcessView installed on it, as Smar ActiveX controls take their security and licensing from this registered ProcessView product.

The steps for configuring DCOM and browsing WebHMI are given below. For further information about DCOM, please refer to the Microsoft Windows help documentation.

Note: If GenBroker TCP/IP channels are used, as they are in WebHMI, neither DCOM settings nor user accounts are necessary. The information in this chapter is provided only as a reference for setting up DCOM. You should not need this information if you use WebHMI.

Configuring DCOM for Windows NT

Creating a User Group

It is necessary to create a group of users who will be able to access WebHMI. This is done either on the Domain Controller or the Workstation, whichever may be the case.

To begin, click the Start button and select Programs > Administrative Tools > User Manager. Select User Manager, as shown in Figure B.1.

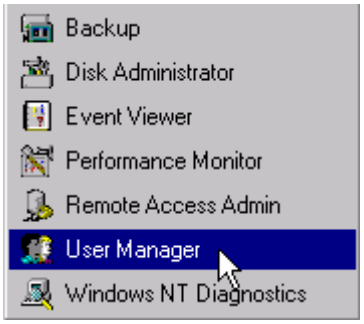


Figure B.1. Opening the User Manager Screen

The **User Manager** screen will appear, as shown in **Figure B.2.** Select **New Local Group** from the **User** menu.

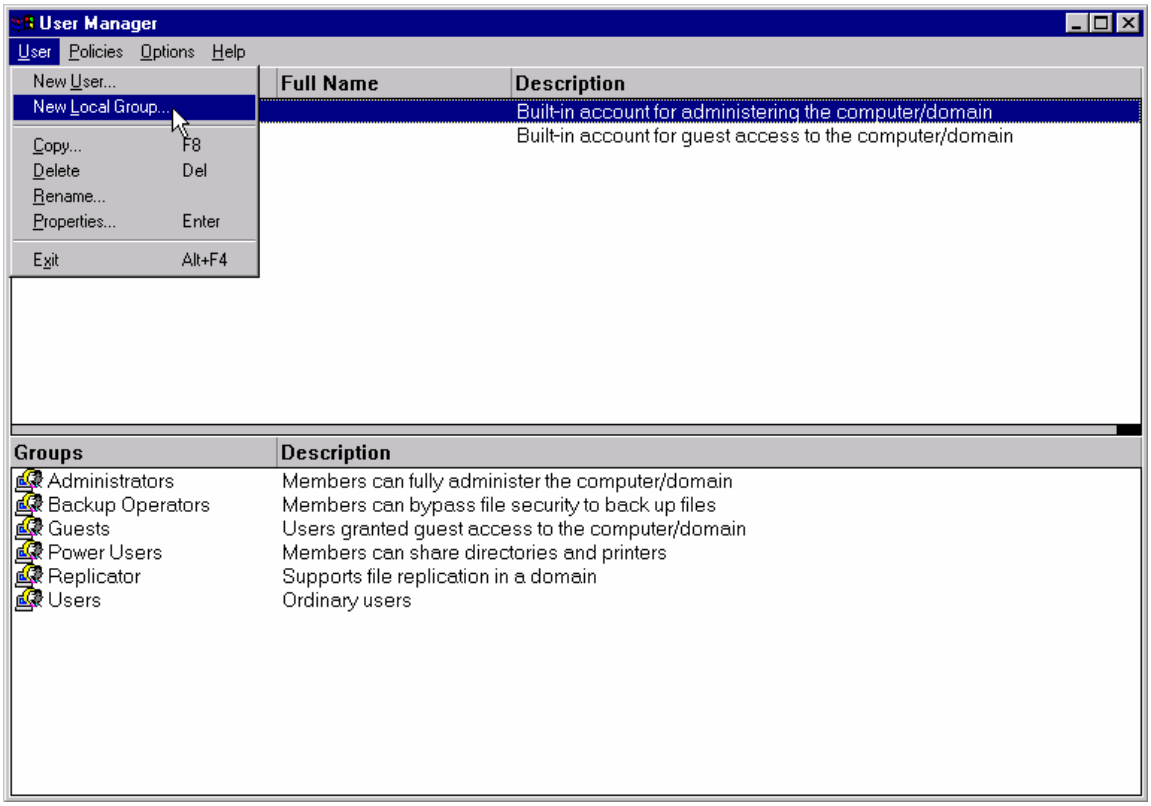


Figure B.2. User Manager Screen

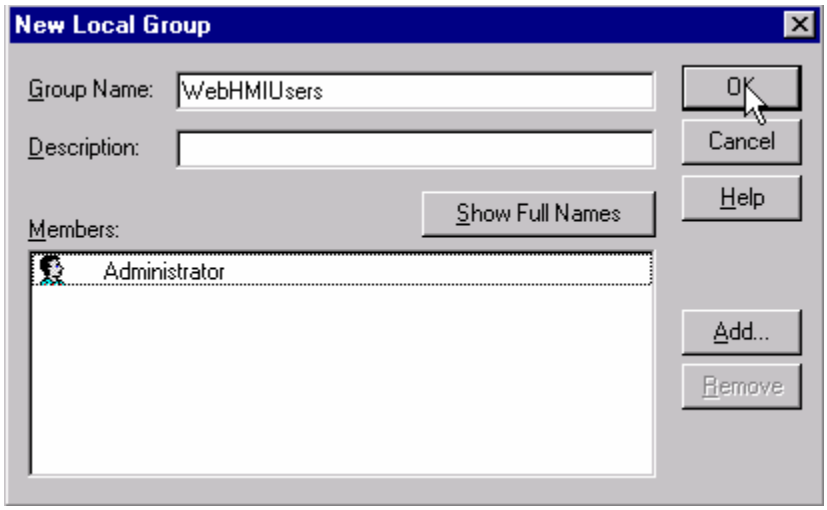


Figure B.3. New Local Group Dialog Box

The **New Local Group** dialog box will appear, as shown in **Figure B.3.** In the **Group Name** field, type "WebHMIUsers" and click the **OK** button. The group name "WebHMIUsers" will appear in the bottom pane of the **User Manager** screen. Double-click the on **WebHMIUsers.** The **Local Group Properties** dialog box will appear, as shown in **Figure B.4.** On the **Local Group Properties** dialog box, click the **Add** button.

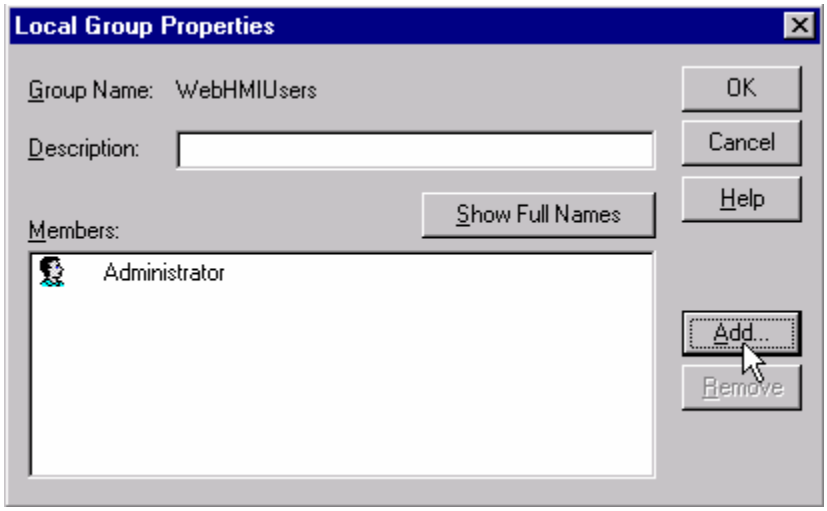


Figure B.4. Local Group Properties Dialog Box

The **Add Users and Groups** dialog box will appear, as shown in **Figure B.5.** In the **Names** field of the **Add Users and Groups** dialog box, select each name that you wish to add to the "WebHMIUsers" group. After selecting the name, click the **Add** button, as shown below.

Note: For the purposes of the example given here, we have chosen **QA_LAB** for the **List Names From** field. You will probably use another source of names when configuring your new user groups.

Each time that you choose a name and click the **Add** button, that name will appear in the **Add Names** pane, as shown below. To find a name that does not appear in the **Names** pane, click the **Search** button, as shown in Figure B.5.

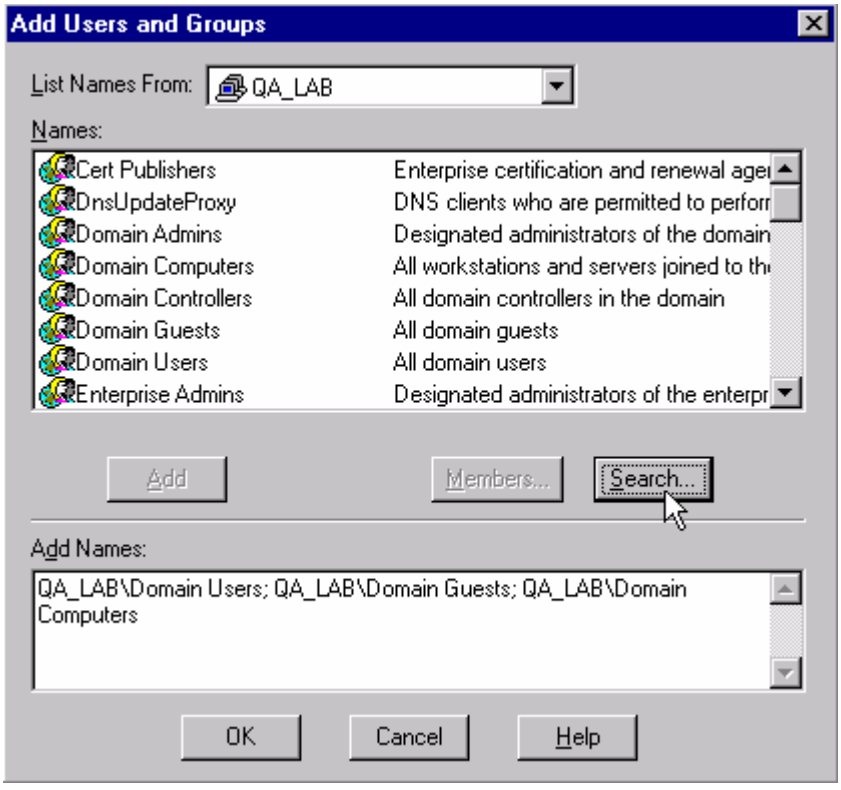


Figure B.5. Names Added to "WebHMIUsers" Group

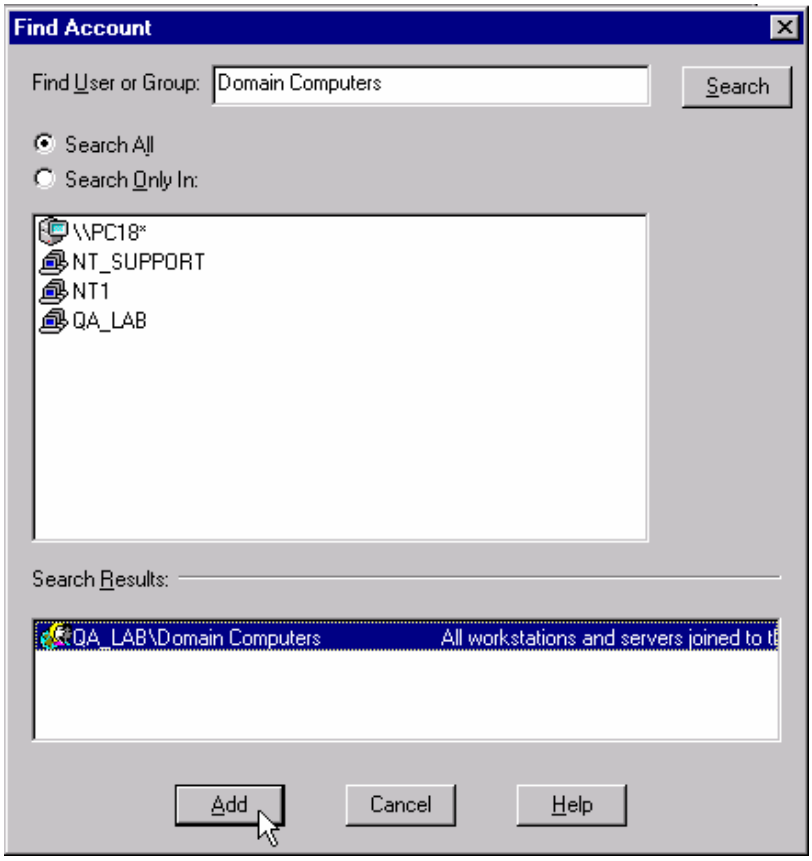


Figure B.6. Add a Found Name to the Add Names Pane

The **Find Account** dialog box will appear, as shown in **Figure B.6.** In the **Find User or Group** field, type the name of the user or group for which you are searching, and then click the **Search** button.

If the name for which you are searching is found, it will appear in the **Search Results** field of the **Find Account** dialog box. Click the **Add** button to place the name in the **Add Names** pane of the **Add Users and Groups** dialog box, as shown in Figure B.6. After you have placed all of the names in the **Add Names** pane, click the **OK** button on the **Add Users and Groups** dialog box**.**

This will cause the **Local Group Properties** dialog box to appear, as shown in **Figure B.7.** Click the **OK** button on the **Local Group Properties** dialog box. This will cause the **User Manager** dialog box to reappear. You have now created the "WebHMIUsers" Group. To conclude this phase of the DCOM setup, select **Exit** from the **User** menu of the User Manager dialog box.

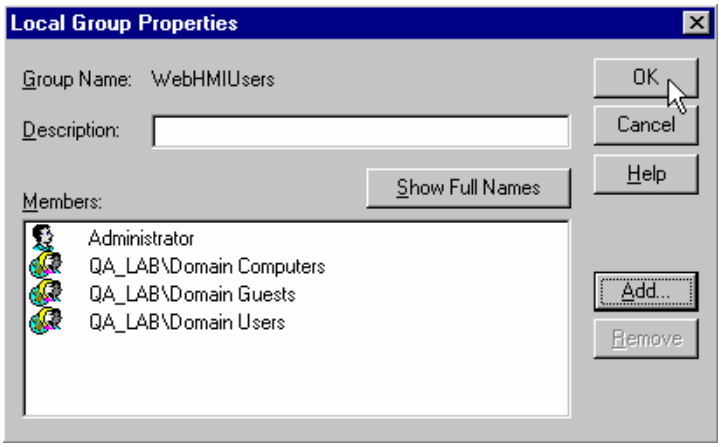


Figure B.7. Local Group Properties Dialog Box

Configuring DCOM Properties

To begin configuration of the DCOM properties, click the **Start** button and select **Run** from the Windows **Start** menu. The **Run** dialog box will appear, as shown in **Figure B.8.** Type "dcomcnfg" in the **Open** field, and then click the **OK** button.

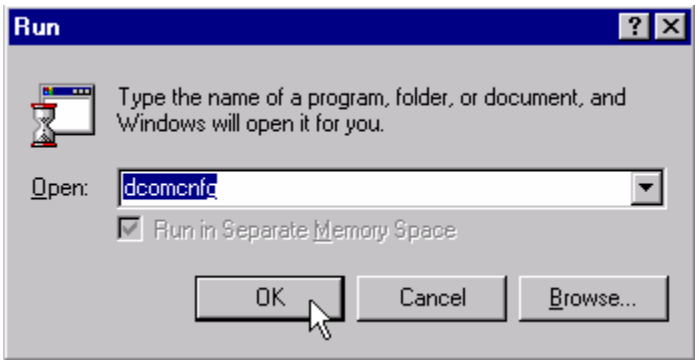


Figure B.8. Opening the dcomcnfg Folder

The Distributed COM Configuration Properties dialog box will open, as shown in Figure B.9. Click on the Default Properties tab of the Distributed COM Configuration Properties dialog box.

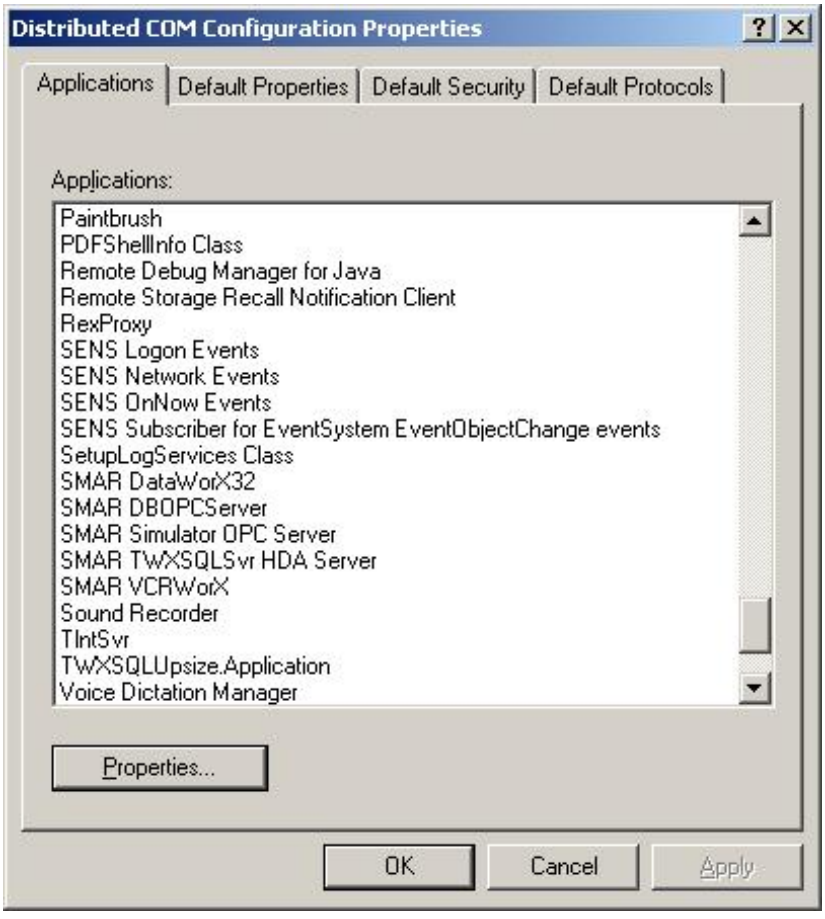


Figure B.9. Distributed COM Configuration Properties Dialog Box

The **Default Properties** tab will appear, as shown in **Figure B.10.** Check **Enable Distributed COM on this computer.** Doing so allows DCOM objects to be created on this machine. Select **Connect** in the **Default Authentication Level** field. Select **Identify** in the **Default Impersonation Level** field. You do this in case security is absent at the object level; the server will then use this level to check permissions in the ACL (Access Control List) of the client machine.

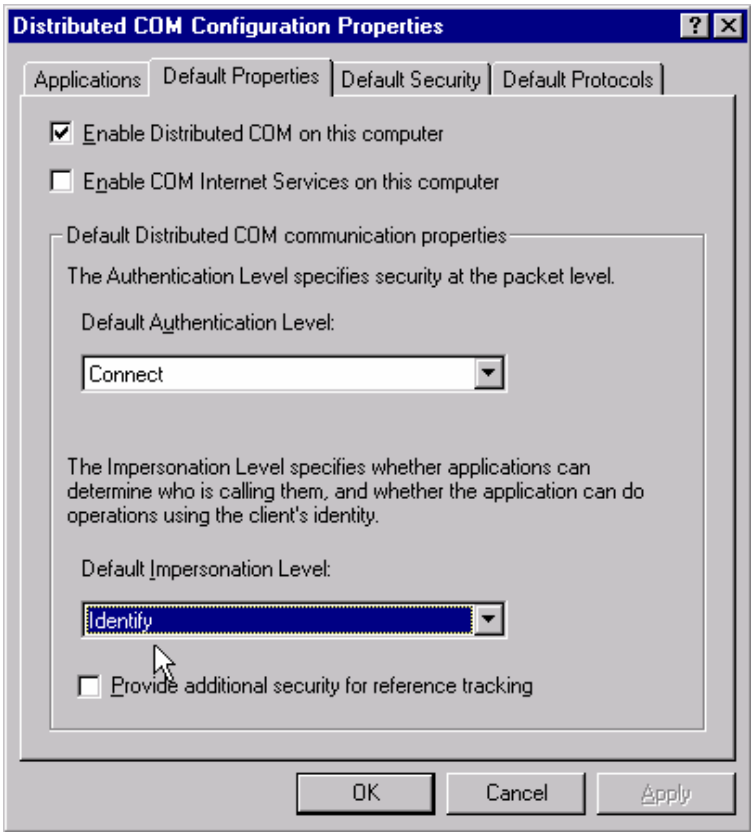

Distributed COM Configuration Properties
Applications
Default Properties
Default Security
Default Protocols
Enable Distributed COM on this computer
Enable COM Internet Services on this computer
Default Distributed COM communication properties
The Authentication Level specifies security at the packet level.
Default Authentication Level:
Connect
The Impersonation Level specifies whether applications can determine who is calling them, and whether the application can do operations using the client's identity.
Default Impersonation Level:
Identify
Provide additional security for reference tracking
OK
Cancel
Apply

Figure B.10. Configuring the Default Properties

Click on the **Default Security** tab to have its page appear, as shown in **Figure B.11.** Click the **Edit Default** button.

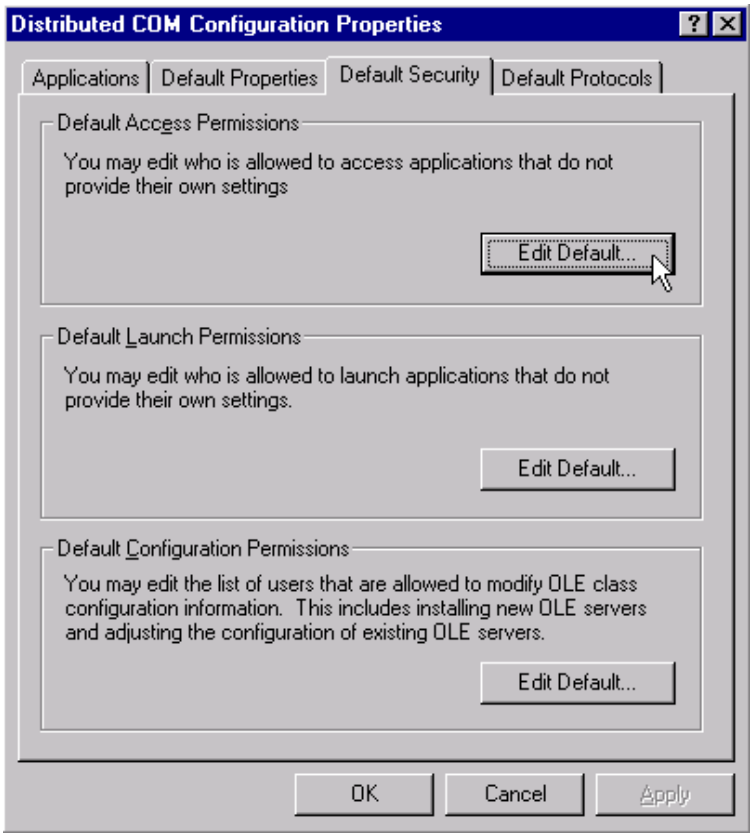

Distributed COM Configuration Properties
Applications
Default Properties
Default Security
Default Protocols
Default Access Permissions
You may edit who is allowed to access applications that do not provide their own settings
Edit Default...
Default Launch Permissions
You may edit who is allowed to launch applications that do not provide their own settings.
Edit Default...
Default Configuration Permissions
You may edit the list of users that are allowed to modify OLE class configuration information. This includes installing new OLE servers and adjusting the configuration of existing OLE servers.
Edit Default...
OK
Cancel
Apply

Figure B.11. Default Security Tab

This opens the **Registry Value Permissions** dialog box, shown in **Figure B.12.** Click the **Add** button.

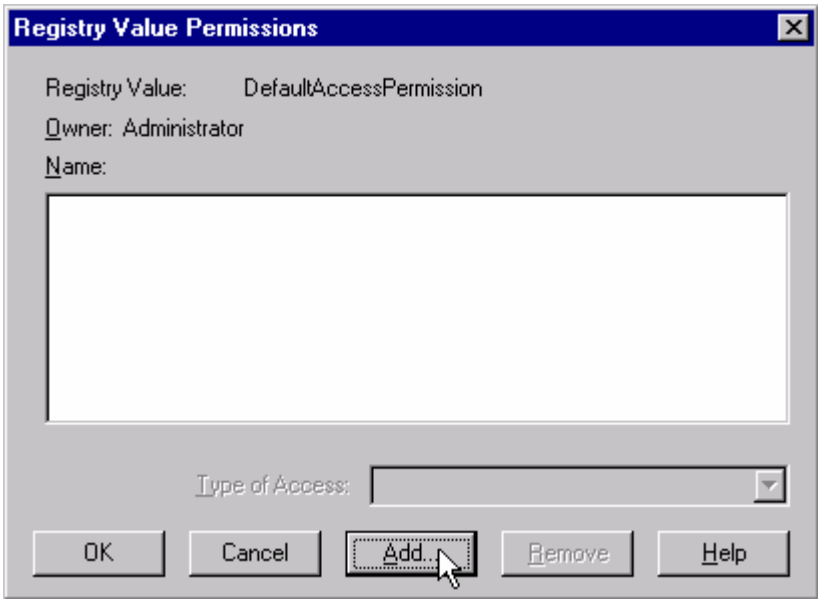


Figure B.12. Registry Value Permissions Dialog Box

This opens the **Add Users and Groups** dialog box, shown in **Figure B.13.** In the **Names** field, scroll down to **WebHMIUsers,** and then click the **Add** button. "WebHMIUsers" will appear in the **Add Names** field, as shown in Figure B.13. With "WebHMIUsers" now in the **Add Names** field and **Allow Access** set in the **Type of Access** field, click the **OK** button.

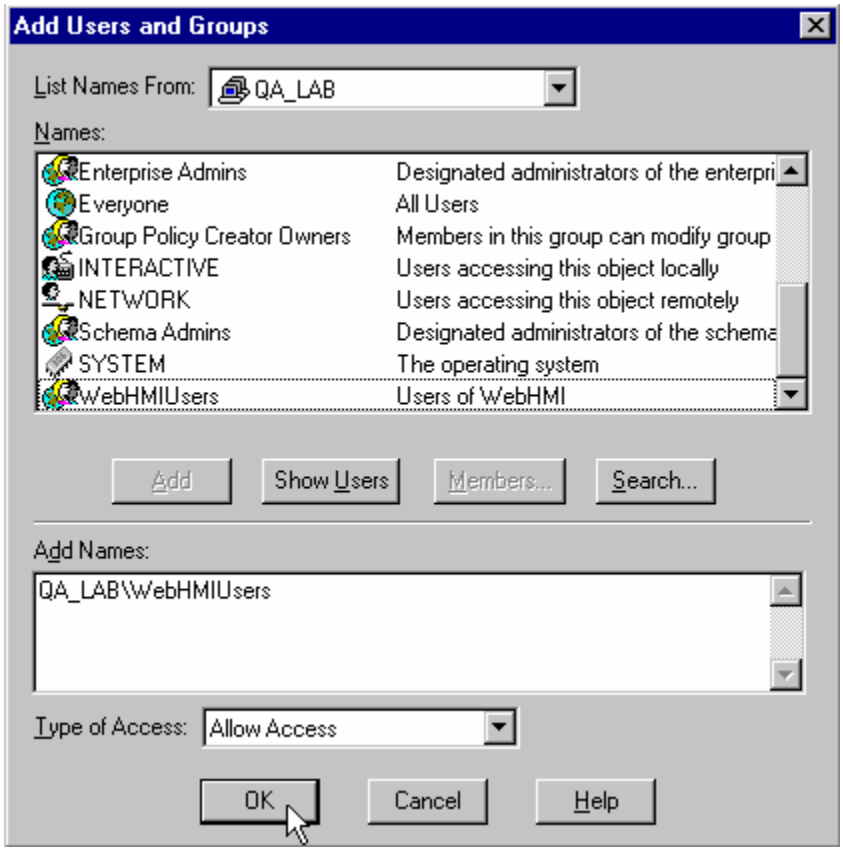


Figure B.13. WebHMIUsers Placed in Add Names Field

The **Registry Value Permissions** dialog box will appear, as shown in **Figure B.14.** With "WebHMIUsers" now in the **Name** field and "Allow Access" set in the **Type of Access** field, click the **OK** button.

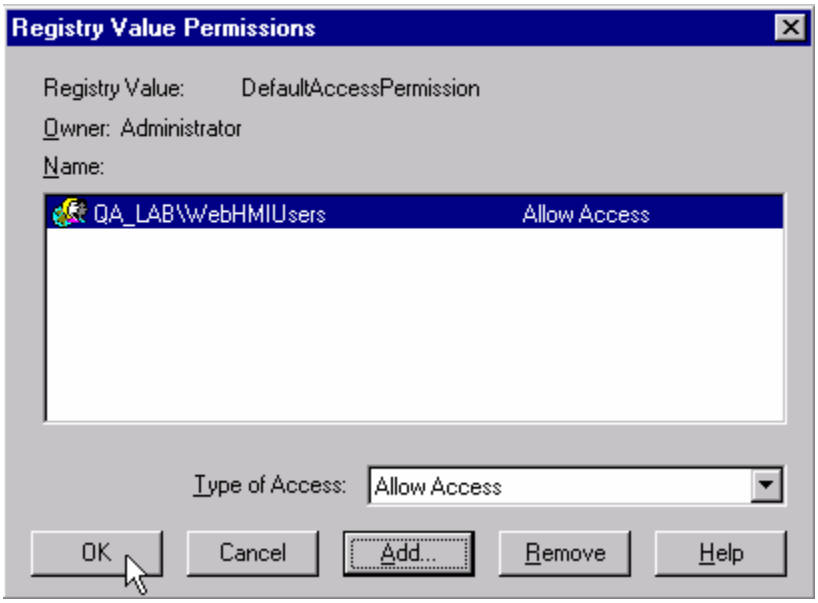


Figure B.14. "WebHMIUsers" Added to Name Field

The Distributed COM Configuration Properties dialog box will appear, as shown in Figure B.15. In the Applications tab, select Security Server, and then click the Properties button.

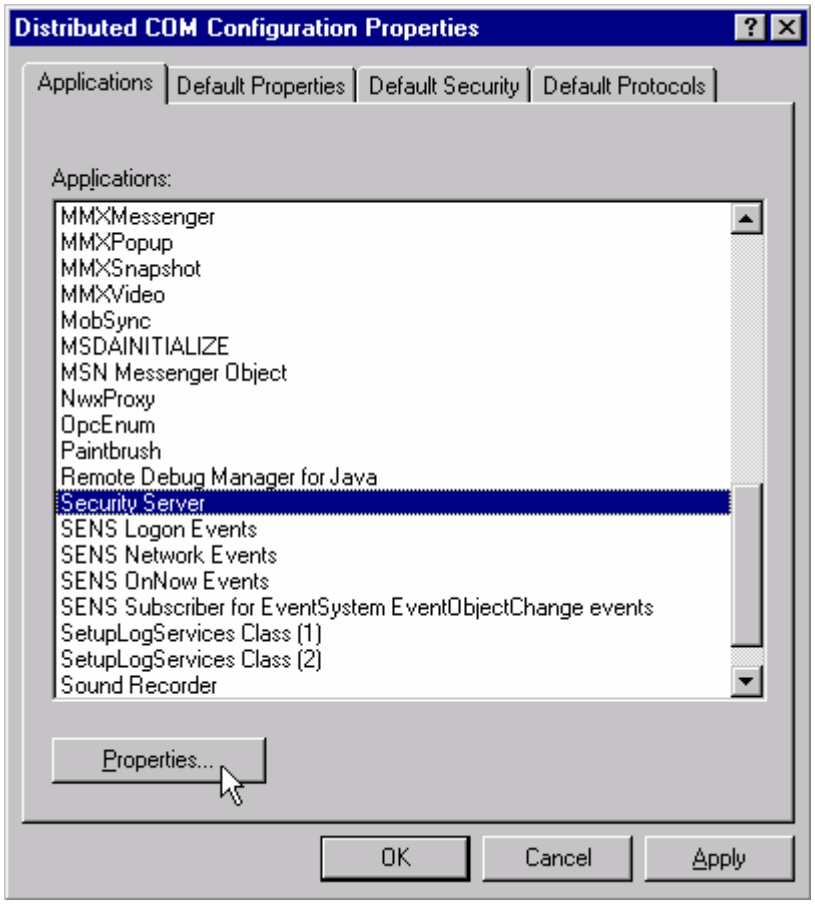


Figure B.15. Setting Properties of Security Server

The **Security Server Properties** dialog box will appear, as shown in **Figure B.16.** On the **General** tab of the **Security Server Properties** dialog box, make sure that the **Authentication Level** field is set to **Default.** Then click on the **Security** tab.

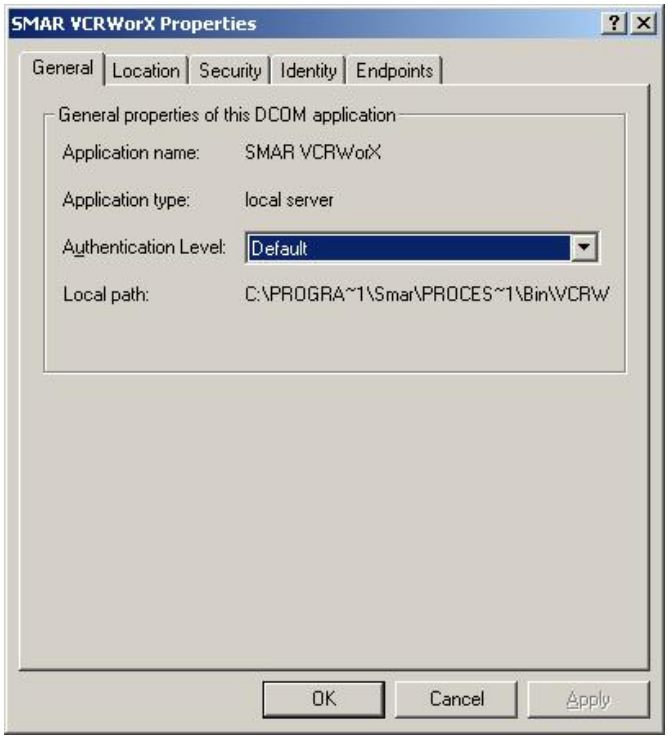


Figure B.16. Security Server Properties Dialog Box

The Security tab page will appear, as shown in Figure B.17. On the Security tab, make sure to select Use default access permissions, Use default launch permissions, and Use custom configuration permissions. Then click on the Location tab.

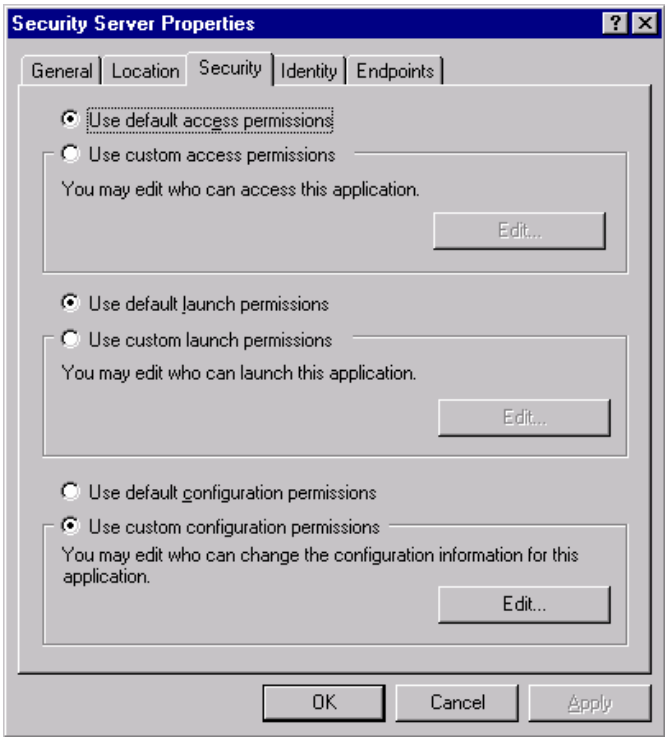


Figure B.17. Setting Permissions

The **Location** tab will appear, as shown in **Figure B.18.** On the **Location** tab, check **Run application on this computer.** Click the **OK** button. The **Distributed COM Configuration Properties** dialog box will reappear. Click the **OK** button.

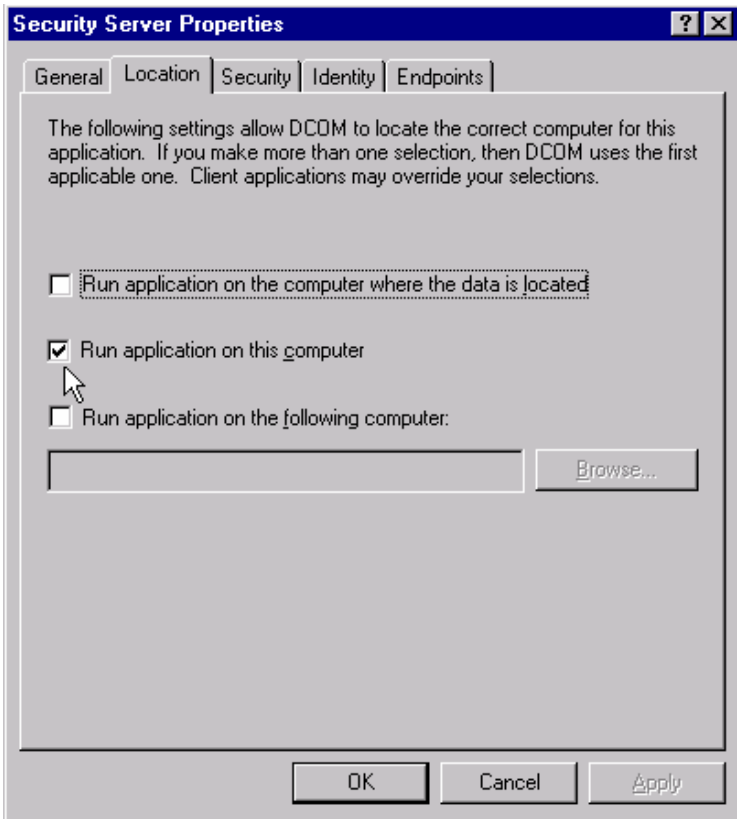

Security Server Properties
General
Location
Security
Identity
Endpoints
The following settings allow DCOM to locate the correct computer for this application. If you make more than one selection, then DCOM uses the first applicable one. Client applications may override your selections.
Run application on the computer where the data is located
Run application on this computer
Run application on the following computer:
Browse...
OK
Cancel
Apply

Figure B.18. Location Settings

You have now completed the DCOM setup process for Windows NT.

Configuring DCOM for Windows 98

Using a Windows 98 Machine As a Web Server

Computers with Windows 98 installed must belong either to a domain or a workgroup. To let this Web server know about all of the permitted users, select **Run** from the **Start** menu, as shown in **Figure B.19.**

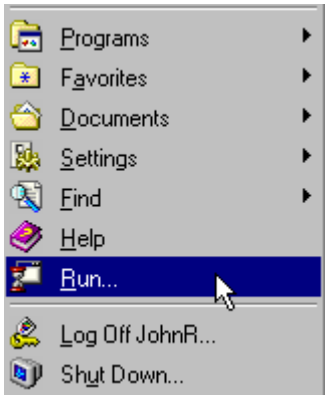


Figure B.19. Opening the Control Panel

The **Run** dialog box will appear, as shown in **Figure B.20.** Type "dcomcnfg" in the **Open** field, and then click the **OK** button.

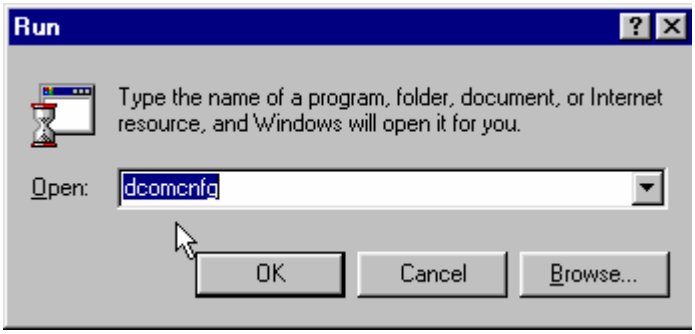


Figure B.20. Run Dialog Box

The Distributed COM Configuration Properties dialog box will appear. Click on the Default Properties tab to open the Default Properties tab page, shown in Figure B.21. Select None in the Default Authentication Level field, and select Identify in the Default Impersonation Level field.

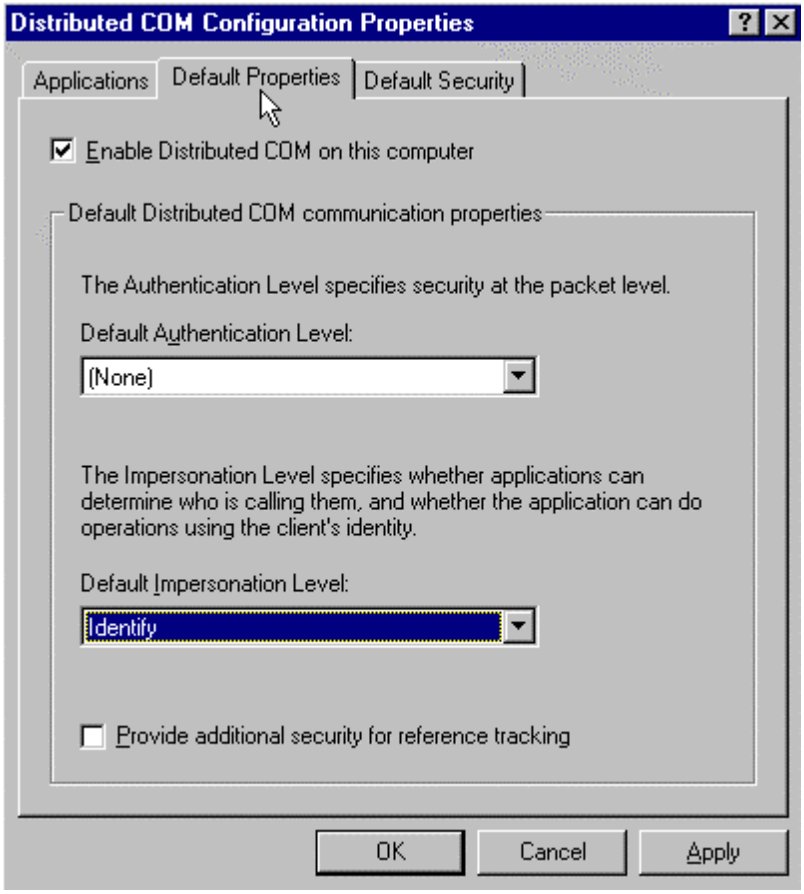

Distributed COM Configuration Properties
Applications
Default Properties
Default Security
Enable Distributed COM on this computer
Default Distributed COM communication properties
The Authentication Level specifies security at the packet level.
Default Authentication Level:
(None)
The Impersonation Level specifies whether applications can determine who is calling them, and whether the application can do operations using the client's identity.
Default Impersonation Level:
Identify
Provide additional security for reference tracking
OK
Cancel
Apply

Figure B.21. DCOM Configuration Properties Dialog Box

Click on the **Default Security** tab, shown in **Figure B.22,** and then click the **Edit Default** button.

Figure B.22. Default Security Tab

The **Access Permissions** dialog box will appear, as shown in **Figure B.23.** Under the name field, select the names of all domain or workgroup users, and then click the **Add** button.

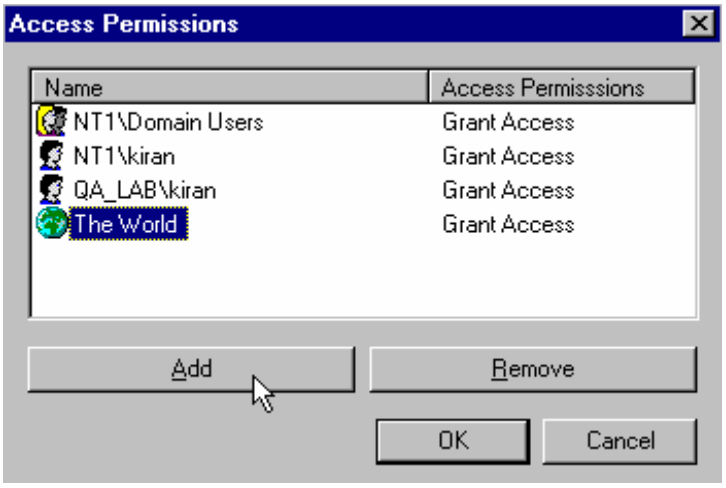


Figure B.23. Access Permissions Dialog Box

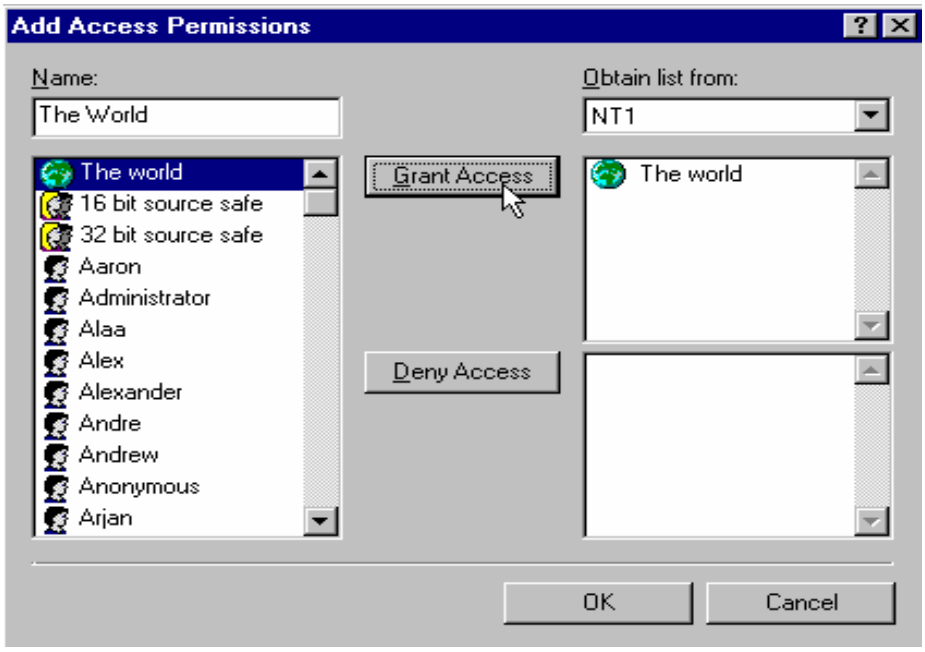


Figure B.24. Add Access Permissions Dialog Box

The **Add Access Permissions** dialog box will appear, as shown in **Figure B.24.** The names that you have selected will appear in the **Name** field and in the right-hand pane. Click the **Grant Access** button, and then click the **OK** button.

The Distributed COM Configuration Properties dialog box will reappear. On the Default Security tab page, check the Enable remote connection check box.

Now click on the **Applications** tab of the **DCOM Configuration Properties** dialog box. In the **Applications** field, select **Security Server,** and then click the **Properties** button, as shown in **Figure B.25.**

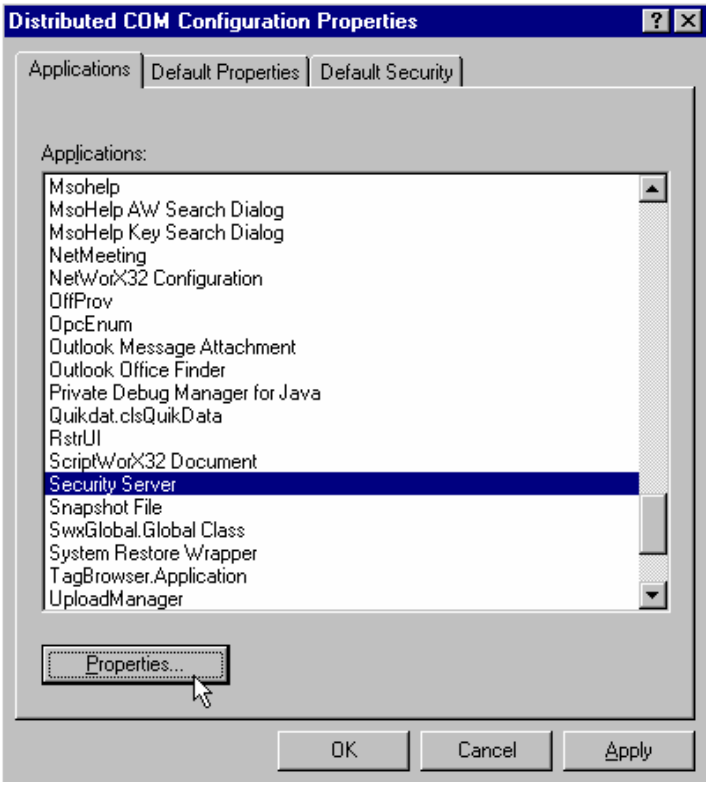


Figure B.25. Selecting Security Server

The Security Server Properties dialog box will appear, as shown in Figure B.26. On the Security tab page, select Use default access permissions.

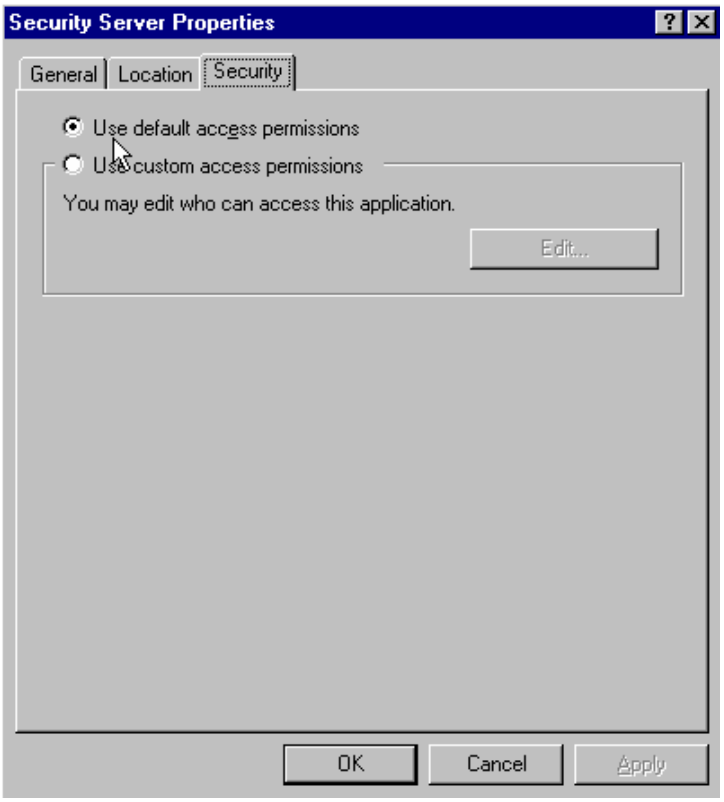


Figure B.26. Security Server Properties

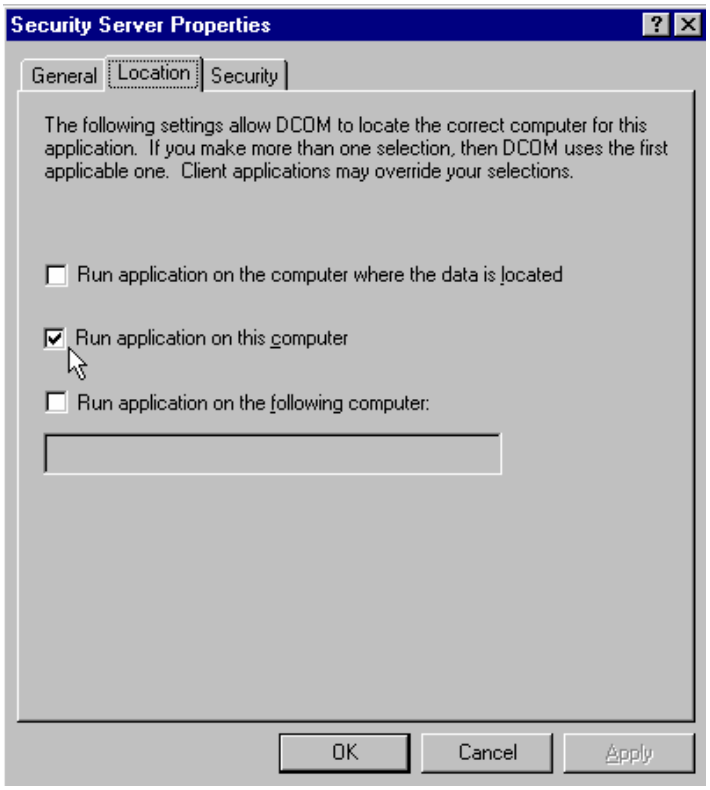

Security Server Properties
General
Location
Security
The following settings allow DCOM to locate the correct computer for this application. If you make more than one selection, then DCOM uses the first applicable one. Client applications may override your selections.
Run application on the computer where the data is located
Run application on this computer
Run application on the following computer:
OK
Cancel
Apply

Figure B.27. Location Tab

As shown in Figure B.27, on the Location tab, check the Run application on this computer option. To return to the Distributed COM Configuration Properties dialog box, click the Apply button.

The **Default Properties** tab page of the **Distributed COM Configuration Properties** dialog box will reappear. Click the **Apply** button.

You have completed DCOM setup for Windows 98.
It is, however, strongly suggested that Windows NT be used as the Web server. If there are two NT machines in a workgroup, then you have to create the same group of users ("WebHMIUsers") on these machines. Alternatively, the same user with the same password should log in at any given time on these two PCs.

If all the computers are in a domain, then the client PCs need not have a list of all users (WebHMIUsers) on them, since the domain controller takes care of granting or denying permissions.

On the Windows 98 client machines, you should do the setup in such a way that they receive their list of users from the domain controller or the NT workstation, whichever is pertinent. To do this, go to **Network Properties > Access Control.** In the **Obtain list of Users and groups from** field, type in the PC name of the Workstation (where the workgroup was created) or the PC name of the PDC (domain controller where the domain was created), whichever is pertinent.

Domains, Workgroups, Rights to the Server Node, and DCOM

DCOM's complexity recommends its use only for intranet and LAN applications.
For WebHMI to function correctly or, more specifically, to establish proper DCOM intranet communication, all client WebHMI PCs must be in the same workgroup or domain as the WebHMI server node.

The DCOM default launch and default configuration settings on the server node must be configured so that everyone logged in on the WebHMI client PC has access to the server node. Any user logged in on a client PC must also have the proper rights to browse the WebHMI server.

For the proper transmittal of information over DCOM between two PCs, these two nodes must be in either the same workgroup or domain.

In any networked application, all of the computers must share common information. From Smar' point of view, the most important function of such applications is **to enable the sharing of user-related information.**

In a workgroup, you have to configure this information individually on each machine. In a domain, one main computer, typically called the Primary Domain Controller (PDC), stores all of this information. When a new PC joins the network:

- In a workgroup, you must configure all of the settings manually.
- In a domain, the PDC will automatically send them to the new PC.